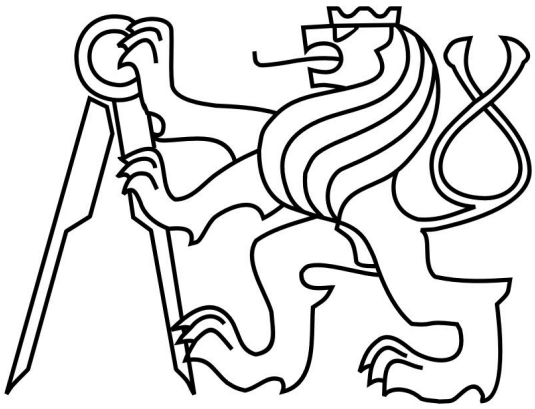
CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

Department of Instrumentation and Control Engineering



Higher Order Neural Unit Adaptive Control and Stability Analysis for Industrial System Applications

Ph.D. Dissertation

Study Branch: Control and Systems Engineering

Author: Ing. Peter Mark Beneš

Academic Year: 2019/2020

Supervisor: doc. Ing. Ivo Bukovský, Ph.D.

Declaration

I hereby declare that this doctoral thesis is my own work and effort written under the guidance of the supervisor doc. Ing. Ivo Bukovský, Ph.D.

All sources and other materials used have been quoted in the list of references along with affiliated project grants for completion of this work.

In Prague on 02.03.2020

.....

.....

signature

Abstract

Given the push in our digitalized industry for advanced, yet comprehensible methods for process identification and control, computational intelligence techniques are readily ongoing in study. Higher-order neural units (HONUs) have proven to be such computationally efficient and comprehensible nonlinear polynomial models for application as standalone process models or as a nonlinear control loop where one recurrent HONU is a plant model and another HONU is as a nonlinear state feedback (neuro)controller (via MRAC scheme). An area which till now has not been so readily studied is their application for real industrial systems whilst also monitoring and ensuring whole dynamical closed control loop stability for HONU-MRAC control loop design, both as offline tuned and online adaptively tuned control loop setups. Alternative approaches as the widely used Lyapunov function, can be used for design of the control law or prove of stability for existing control laws about an equilibrium or given point in state-space. However, in practical engineering applications such methods although proving stability about an equilibrium point are not always fitting to the design goal of achieving optimal tracking performance but rather stabilization. Furthermore, if not also proven to be bounded-input-bounded-output/state (BIBO/BIBS) stable with respect to the control inputs, such designs can even lead to poor performance and damage. Therefore, the main contribution of this dissertation is to introduce two novel real-time BIBO/BIBS based stability evaluation methods for HONUs and for their closed control loops. The proposed methods being derived from the core polynomial architectures of HONUs themselves provides a straightforward and comprehensible framework for stability monitoring that can be applied to other forms of recurrent polynomial neural networks. Both methods are comprehensively analysed and validated through various nonlinear system examples as well as new results presented from the rail automation field for real time industrial process control via HONU-MRAC design. Further directions are also highlighted for sliding mode design via HONUs and multi-layered HONU feedback control presented as a framework for low to moderately nonlinear systems.

Acknowledgements

The author of this work would like to acknowledge several key people who have invested a significant amount of time and support into seeing through the completion of this research. I would firstly like to acknowledge my parents Ing. Jan Beneš and Anita Benesh, as well as brother Ing. John Beneš for their extensive love and support throughout my studies at the Czech Technical University in Prague. I further express my gratitude for their ongoing encouragement, love and support to always strive to achieve the most of your potentials both on a professional and personal level.

I would further like to thank my supervisor doc. Ivo Bukovský, Ph.D. for investing his time and efforts into my research and completion of my study program and further this dissertation work. Further I would like to acknowledge Prof. RNDr. Sergey Čelikovský CSc. for his valuable advice and time for consulation during the preparation of this work. In addition, I would also like to acknowledge the following study grant SGS12/177/OHK2/3T/12 "Non-conventional and cognitive methods of dynamic system signal processing" for support during this work. The Technology Agency of Czech Republic Project No: TE01020038 "Competence Centre of Railway Vehicles" and the EU Operational Programme Research, Development and Education and from the Center of Advanced Aerospace Technology CZ.02.1.01/0.0/0.0/16_019/0000826. In addition, to Siemens s.r.o for their tremendous help and support in providing me with the means for completion of this work.

Ing. Peter Mark Beneš
(Author)

Nomenclature

$\bar{\mathbf{A}}(k);\hat{\mathbf{A}}(k)$	… matrix of dynamics for HONU model (for DHS and DDHS resp.)
$\bar{a}_i,\hat{a}_i$	… i^{th} coefficient of recurrent HONU matrix of dynamics (for DHS; DDHS resp.)
$\bar{\alpha}_i,\hat{\alpha}_i$	… i^{th} coefficient of recurrent HONU-MRAC matrix of dynamics (for DHS; DDHS resp.)
$\bar{\mathbf{B}}(k);\hat{\mathbf{B}}(k)$	… matrix of inputs for HONU model of (for DHS and DDHS resp.)
$\bar{b}_i,\hat{b}_i$	… i^{th} coefficient of HONU input matrix, (for DHS; DDHS resp.)
$\hat{\beta}_i$	… i^{th} coefficient of HONU-MRAC input matrix for DDHS
$C_i(.)$	… sum of coefficients corresponding to state variable term $\hat{x}_i$ (DDHS)
$\mathbf{colx}^r;\mathbf{col}\xi^r$	… long-column vector of up to r^{th} or γ^{th} polynomial order terms. (**x**…plant inputs; ξ …controller inputs)
$\bar{\mathbf{M}}(k);\hat{\mathbf{M}}(k)$	… matrix of dynamics for HONU-MRAC control loop (for DHS and DDHS resp.)
$\bar{\mathbf{N}}(k);\hat{\mathbf{N}}(k)$	… matrix of inputs corresponding HONU-MRAC control loop (for DHS and DDHS resp.)
$n_x;n_\xi$	… length of neural input vector **x** and ξ respectively
$n_u;n_y$	… lengths of step-delayed history u or y in neural input vectors.
p	… adaptive HONU controller output gain
$r;\gamma$	… order of standalone HONU model and HONU feedback controller resp.
$u;d$	… process control input; desired value (set point)
$\mathbf{w};\mathbf{v}$	… long-vector representations of neural weights of HONU (**w**…plant; **v**…controller)
$\Delta\mathbf{w};\Delta\mathbf{v}$	… neural weight updates of HONU (plant; controller)
$\bar{\mathbf{x}}$	… affine control generalized state vector (for DHS)
$\hat{\mathbf{x}}$	… sub-polynomial decomposed state vector (DDHS)
y	… real process output
$\tilde{y}$	… neural output from HONU (plant)

List of Abbreviations

BIBO	...bounded-input-bounded-output
BIBS	...bounded-input-bounded-state
BIBS-RHS	...right-hand side
BPTT	...back-propagation through time
CNU	...cubic neural unit of third polynomial order (i.e. $r=3$ ...model, $\gamma=3$ ...as a controller)
DDHS	...discrete-time decomposed HONU stability condition (method developed by author)
DHS	...discrete-time HONU stability condition (method developed by author)
GAS	...global asymptotic stability
GD	...gradient descent
HONU	...higher-order neural unit
HONU-MRAC	…closed control loop where one HONU is a plant model and the second HONU is a feedback controller
ISS	...input-to-state-stability
LM	...Levenberg-Marquardt algorithm
LMD	...local matrix of dynamics
LNU	...linear neural unit of first polynomial order (i.e. $r=1$ ...model, $\gamma=1$ ...as a controller)
MRAC	...model reference adaptive control
NN	...neural network
QNU	...quadratic neural unit of second polynomial order (i.e. $r=2$ ...model, $\gamma=2$ ...as a controller)
RLS	...recursive least squares algorithm
RR	...roller rig (experimental railway stand)
RTRL	...real-time recurrent learning

CONTENTS

1 Introduction

As our modern industry undergoes a push towards digitalization, optimisation and increased efficiency is growing to not only become a desirable feature but a necessity within the industry, largely due to factors associated with increased production rates, technological changes and further demands for increase in flexibility [1]. With this the demand for computationally advanced, yet comprehensible process identification and control methods which the engineering personal at hand can maintain is growing. Though many advanced methods have been proposed in the field of adaptive control their applicability for smaller scale industrial processes can be quite over-engineered or too complex especially where a majority of practical industrial processes are of linear or low to moderately non-linear process dynamics. Therefore, one motive of this dissertation is to present such computationally advanced and comprehensible package for process identification, control and stability monitoring of real industrial systems, via higher-order-neural-unit model reference based control (HONU-MRAC). As this dissertation is focussed on low to moderately nonlinear process dynamics, the aim of this section is to summarize existing state-of-the-art process identification and control approaches and setup the direction for deeper study of computational intelligence based methods and adaptive polynomial neural network based architectures as the main topic of this work.

Till date, the Czech Technical University in Prague (CTU) has published numerous works which are focussed on more conventional, yet advanced approaches to nonlinear process control. As a challenge with control of any nonlinear process lies often in the identification of the process, in [2], [3] an approach to multiple-input-multiple-output (MIMO) process modelling is presented via LOLIMOT method. The approach is based on generating approximation of any function in parameters by locally linear functions and globally rational functions which can be switched to provide continuous form of grid data approximation. With regards to control, in the work [4] a input-shaping technique is investigated for non-vibrational control of flexible mechatronic systems. There, an optimised precomputed control curve in each control point is found at first, followed by application of an online input shaper to transform an arbitrary input signal to a non-vibrational form. A further technique for control of flexible mechatronic system is via wave-based technique. In [5] a wave model is composed of the mechanical system, where via derivation of corresponding launch and reflection transfer functions which are used to compose the respective wave-control. A further innovation from the department of Mechanics, Biomechanics and Mechatronics is an

extension on Linear Quadratic Regulators (LQR), a Nonlinear Quadratic Regulator (NQR) is presented in the work [3], [6] for control of nonlinear multi-body systems. This method also referred to as State-Dependant Riccati Equation (SDRE) strategy [7], serves as an effective algorithm for capturing nonlinearities of the system dynamics via linear structure having state dependant coefficient matrices (SDC). These SDCs are then used for solving algebraic Riccatti equations online to give the control law.

Apart from the reviewed conventional forms of nonlinear process identification and control, a large push in current date research is towards computational intelligence methods as such that of fuzzy logic or neural network forms [8]–[12] and their collaboration as exampled in [13] & [14]. Initially founded by P. V. Kokotovic et al. [15], till this day a large focus in research is based on adaptive back-stepping design. Extensive studies have been published regarding their applicability towards stabilisation of stochastic nonlinear systems [16], [17]. However due to the challenges of requiring precise description of the applied nonlinear systems, an enhancement in adaptive back-stepping techniques may be found in the use of fuzzy logic. Relevant examples include [18], [19] as well studies focused towards its application with neural networks. Advanced control algorithms with neural networks feature backstepping control [20], [21] which are quite competitive for non-linear dynamic processes. However, from these works though the presented methods are advanced the key driver for such forms is the transformation of higher order system description to a lower order, or set of lower order problems especially useful for multiple-input-multiple-output systems. As in the sense of backstepping control, a systematic approach can be found in constructing a Lyapunov function based control law to ensure stability of the negative derivative in every step. Though advantageous due to the cancellation of cross-coupling terms, such methods have more emphasis on the stable control law as opposed to the ensured of control performance itself. For practical industrial applications with moderate non-linearity, an adequately advanced and comprehensible control form is Model Reference Adaptive Control (MRAC). Extensive studies have been presented for MRAC control loop design, focussing not only on new testing methods for controller parameter adjustment, but also providing techniques for specialised engineering applications. An advantage is via a data driven approach, to force unknown systems to track defined system dynamics [22]. The works [23] & [24] for example present a model reference based form of adaptive control utilising an adaptive update rule based on Lyapunov function based criterion for adjustment of its respective controller parameters, which in lieu is used for either direct or indirect controller parameter tuning. Due to recent orientation of research being aimed towards soft

computing, the incorporation of fuzzy logic [25] & [26] and neural network based methods [27], [28] & [29] for MRAC showcases their increased suitability for control of systems with nonlinear properties. A different and promising area of adaptive control for nonlinear systems is that of dynamic programming and optimal control based techniques used for controller parameter adjustment, either in a direct or indirect manner. A neural network form of adaptive control via dynamic programming techniques may be found in the works by D. Liu and there in [30], [31].

A further discipline of adaptive control which is trending in studies in our modern digitalized industry, is that of polynomial function based neural networks (PNNs) and as a further subclass higher order neural networks (HONNs) for adaptive identification and control of dynamic systems. Studies in higher order neural units (HONUs) control [32], [33] & [34] have proven such advanced capability for identification and control of even moderately non-linear process dynamics whilst being comparable in computational complexity to conventional linear control forms. Earlier works [35] & [36] showcase the applicability of PNN based architectures for dynamic system identification of HONUs which as presented in [33] & [34] are a rather novel subclass of HONNs which are shown to deliver powerful performance as an approximator in terms of rate of convergence to the desired behaviour of linear systems along with computationally efficient training of unknown neural weights in comparison to implemented MLP based neural network forms [33]. Due to the complexities in model reference control design in building a stable control law, whilst ensuring convergence to the reference model, an advantageous approach is the design of a standalone HONU adaptive model with extended HONU in state-feedback. This form of adaptive control loop design falling under the umbrella of MRAC design is termed more directly as a HONU-MRAC closed control loop. Till now HONU-MRAC closed loop control has not been so readily presented nor extensively published, especially for real-time process control on physical industrial systems. And is hence, one of the key objectives behind this dissertation work and the produced publications following. In contrast to conventional forms of NN based adaptive control architectures [37], [38], their computationally efficient structure due to in-parameter linearity and customizable nonlinear approximation strength via the polynomial order of HONU, exhibit strong advantages for real-time application to industrial process control, as seen further in section 5 of this dissertation.

With every innovation in adaptive control also comes the challenge of guaranteeing stability of the proposed adaptive model as an approximator and further adaptive control loop as a

whole. In [39], a recent study was presented for guaranteeing convergence of the gradient descent based supervised learning algorithm as applied to HONU adaptive models. However, due to HONU-MRAC control loop design not being so readily published, a rather uncovered area of study for this control approach is the stability evaluation of a HONU adaptive model following adaptive identification as applied to new process data and further, the guarantee of whole HONU-MRAC closed control loop stability both in the sense of a constant parameter controller and as an online adaptive control loop. Till now, several key methods as applied to NN forms of reference model based adaptive control loops include [37] & [40] where a more traditional approach of constructing the adaptive control law via a suitable Lyapunov stability function is exampled. Further, relevant approaches are related to recurrent neural networks (RNNs), a similar approach based on RNNs may be found in [41] focussed on nonlinear systems with bounded nonlinearities. Another readily published method in the field of RNN adaptive control, is the Linear Matrix Inequality (LMI) technique. In [42] & [43] both constant and time-varying delayed neural networks are investigated via an LMI approach in lieu with the Lyapunov-Krasovskii theory. However, with regards to practical engineering applications it is often more advantageous to analyse boundedness of the process states with respect to given bounded process inputs, as although proof of a stable equilibrium point or specific state-point may be justified, bad performances or large damage can still occur. Therefore another readily studied approach is that of BIBS and as a more universal definition ISS stability [44], [45]. Examples of its application to RNNs include [46], where two algebraic criterions for verification of ISS are developed for a class of time-varying delay RNNs. Further, in [47] an exponential ISS verification is exampled for a class of multiple time-varying delay RNNs. As the above mentioned works cover several key stability analysis methods in the realm of RNNs, a key work to mention is [48] and references there in. In their work, comprehensive review of the major approaches for stability evaluation of RNNs is presented. Following comprehensive study, a natural conclusion was drawn that there is not one universally best method for stability evaluation and that each method features its strengths with regards to the adaptive control loop design at hand.

Following review of the above works, it can be stated that although advanced algorithms do exist and are quite well theoretically proven for neural network based control, there is still a gap in physical implementation within real industry. Moreover, the maintainability for engineering personal at hand to comprehend and troubleshoot such algorithms during real application. For this reason, polynomial neural network based approaches are trending for practical use in our digitalized modern industry. An added advantage of the HONU-MRAC

approach as exampled in chapter 8.5 is the ability to apply control design as a constant parameter loop online which is often a requirement of safety-related industrial applications. Furthermore, a comprehendible online stability monitoring of the whole applied control loop which may be applicable for both adaptive and constant parameter polynomial architectures remains to be a necessary yet open topic, especially for HONU-MRAC based design. Therefore, this dissertation aims to advance the theories of HONU adaptive identification and adaptive control for moderately non-linear single-input-single-output (SISO) engineering processes. Several new remarks are proposed regarding the use of the recursive least squares algorithm and a derivation of an adaptive feedback gain parameter for HONUs with their nonlinear closed control loop. In chapter 10, future directions of research are proposed via the introduction of a HONU sliding mode control architecture utilizing the derived HONU decomposition introduced in this dissertation and extension of the decomposition approach to similar polynomial neural based architectures.

Following this, the major contribution of this dissertation is to investigate HONU model and HONU-MRAC closed control loop stability, which can also be extended to similar polynomial neural network based control architectures. Due, to the property of HONU models featuring a customable quality of nonlinear approximation via polynomial order while their optimization is a linear problem due to their linearity in parameters derivation of two novel stability evaluation methods are presented in this dissertation. The first (DHS) (Section 5) backbones from the principles of nonlinear system linearization, for evaluation of local BIBO stability in vicinity of the evaluated state points in state space form. The functionality of this approach is then presented in section 8. The derived DHS approach also serves as one form of validation for the second presented approach (DDHS). The DDHS method, as also inspired from concepts of reduced order modelling as presented via LOLIMOT technique and NQR (or SDRE approach) [2], [3] results in an intrinsic decomposition of a nonlinear HONU structure into sub-polynomial state space form (Section 7). The approach ultimately yields a time-variant state-space form where extended with the principles of input-to-state (ISS) stability, a BIBS conclusion of stability can be evaluated across all samples in time for the monitoring whether the neural model and further whole HONU-MRAC control loop maintains dynamical stability along its path in state space for actual process inputs. Following the derivation of the DDHS theories, a stricter condition is developed termed as DDHS(Strict) which more clearly pronounces an onset of unstable process dynamics due to the applied HONU or HONU-MRAC control loop as compared with the investigated Lyapunov function based approach in section 8.4. As the DDHS and

DDHS(Strict) are state transition based approached derived from the model dynamics itself, it serves as a more comprehensible and advantageous approach to local stability criteria which do not always account for neighbouring unstable dynamics in time. The newly proposed approaches are investigated not only on several theoretical nonlinear dynamic system examples, but also on real industrial systems from the rail automation field to justify its use and applicability for real-time industrial process control.

2 Overview of the Methods and Problems in Adaptive Control

Though the field of adaptive control remains to be vast, we may categorise three key areas of control objectives for which the reviewed forms fall underneath:

- *Control input adjustment methods*: A readily published form of such control is to derive a Lyapunov control function based law for newly applied input signals to the dynamical system. Such approach relies on the justified stability of the control law via consideration of the negative derivate in each time point of the Lyapunov control function. Though such methods are quite readily employed in control design, less emphasis is pushed on maintaining a tracking (desired) signal and more focus is pushed towards stability of the control law. Another popular form is Model Predictive Control (MPC) [49]–[51] where the main objective is to calculate the future control signals in the scope of a prediction horizon, such to optimise a performance criterion. This criterion usually contains the control effort and is aimed to minimise the error between the predicted process outputs based on an explicit use of a model and the reference behaviour. Other advanced forms include adaptive backstepping design [20] and a quite trending technique of sliding mode control [52], [53]
- *Heuristic tuning based methods*: A popular form of such control is Adaptive Dynamic Programming (ADP) [54], [55] which may be divided into the basic structures of heuristic dynamic programming (HDP) as exampled in the works [31] [30] and Dual heuristic programming (DHP). HDP remains till now amongst the most popular form of ADP where the main goal is to estimate a cost function J for a given policy. Often such design incorporates an action and model network to map the process environment. Based on minimisation of an error measure over time a penalisation or reward is calculated to compute the output of the critic network which is an estimate of the cost function J. In DHP rather than focussing on just the cost function estimation, it is the gradient of a cost function that is estimated.

- *Parameter adjustment based control*: More conventional forms of such control are adaptive PID controller tuning or Model Reference Adaptive Control (MRAC), which primarily is focussed on calculation of the most optimal controller parameters for minimisation of the process error in relation to a reference model or set point value. Till this day more emphasis is pushed towards variations with use of computational intelligence methods as such neural-networks and fuzzy based methods. Parameter adjustment forms may also be combined with input adjustment forms, an example of such approach is the Model-Reference Sliding Mode Control as reviewed in [56].

2.1 Closed Loop Feedback Methods of Adaptive Control for SISO Processes

An active area of research in the field of adaptive control is that of Model Reference Adaptive Control (MRAC). MRAC is a closed control loop method consisting of a reference model and inner control loop consisting of the process itself and applied controller in an outer feedback loop as a controller parameter adjustment mechanism to minimise the difference between the applied reference model and plant output. In this current time, more emphasis in research has been pushed towards utilising soft computing techniques such as fuzzy and neural network based methods of MRAC adaptive control, however research in MRAC design with variable structures and advanced controller adjustment technique, whether they be direct or indirectly applied to the process are readily being extended to this field. For the scope of this research, a primary focus will be into computational intelligence forms of MRAC with mention to various state of the art variations of MRAC adaptive control. A classical example can be seen in Figure 1 from [56].

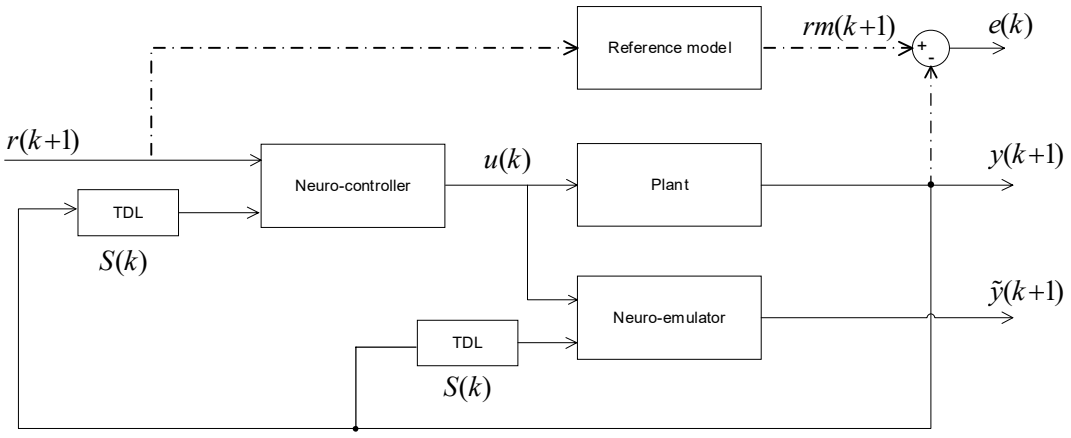


Figure 1 – Structural scheme of Model Reference Adaptive Neurocontrol as presented in [56]

An initial work to mention may be found in H. Qiang [23]. There, an MRAC control loop is presented where an adaptive control law construct via the previous controller output, previous process input and furthermore, a control error as a difference between a defined reference model and plant output is presented. This adaptive law serves to recalculate the parameter coefficients for tuning of the applied controller on the engineering process. In this form, the author's design of the adaptive control law is dictated by a Lyapunov based function, which under simulation of a linear time invariant system, yields a mean control error peak of 0.1 with quite desirable adhesion of the system output to the desired behaviour of the process. The control effect yields that the employed adaptive control law is still adequate even when a significant change in process parameters occur. However, further experimentation is still required to justify the presented adaptive control algorithm for practical implementation on real engineering processes. A further example of such Lyapunov based adaptive control law via MRAC can be found in T.H Liu et al. [24]. There a direct method of adaptive control is employed, with the output of the adaptive control law being the direct controller parameters for the applied controller to the system (linear motor), manipulated via an inverter with a PC based control algorithm. The results in comparison with a back-stepping based controller yields similar adhesion to the desired behaviour of position and speed for the linear motor. Particularly on application to desired square wave set and sinusoidal wave set responses. The authors of this publication compare their results via an integration of the square of position error (in centimetres). Following tests of the presented controller architectures under a square-wave set of desired behaviour the back-stepping controller yields 0.003801[cm] whilst the MRAC achieved 0.004646[cm]. In the sense of the sinusoidal wave set here the MRAC outperforms the back-stepping controller via a deviation of 0.000906[cm]. Further testing on the presented square waves set under different load disturbances applied to the linear motor yield that the MRAC based controller performs with the best adhesion to the desired square wave set behaviour with a significantly larger deviation in favour of the MRAC based controller.

H.D. Patino et. al. [28] (Figure 2) present a different mechanism behind the concept of model reference adaptive control (MRAC) being that of the use of a neural network controller, in which the neural network controller (neuro-controller), is adaptively updated via an error calculation in the difference between the real plant output and output of the reference model and Lyapunov theory in lieu with a sigma modification update rule. The presented neuro-controller may be either implemented via a radial basis function network (RBF) or feed-forward neural network. From this work two key results are exhibited, the first is via a case

study on a nonlinear model of a ship steering system, here during the first 150 seconds of simulation the mean square error of the employed algorithm is reduced to 0.05, implying excellent adaptation of the controller parameters. On comparison with the desired output of the process the curves representing the desired form of control and that of the actual controlled plant seem to coincide justifying the low square error and suitability for control of this system. In the second case study an RBF neural network architecture was used on a theoretical nonlinear plant. Here over 50 seconds, the mean square error is significantly reduced to 0.08, with the output of the controlled plant itself following closely to the desired behaviour apart from the initial 3 seconds of the simulation, where various degrees of overshooting can be seen. From these theoretical case studies, potentials in use of this format for control of nonlinear systems is evident. However as underlined by the authors, further study with regards to the robustness of the employed RBF neural network based method along with experimentation on real engineering processes is necessary.

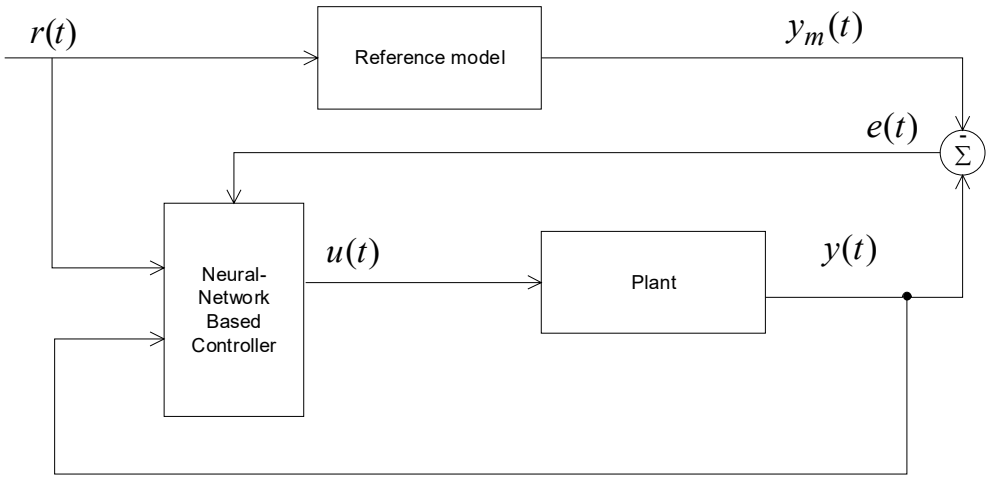


Figure 2- Neural network-based model reference adaptive control system structure as presented in [28]

Another more recent example of MRAC adaptive control for application to a ship steering system may be found in the work [57] by Y. Yang et al. In this work a fuzzy logic based MRAC scheme is employed. An advantage of using fuzzy based methods is the conversion of the control parameters from a more quantitatively based approach into that of a more qualitatively based approach. Although the performance of such mechanism is dependent on the construction of fuzzification and the defined rule basis along with appropriately chosen defuzzification methods, many works have proven its worth for control applications, particularly in the field of uncertain nonlinear systems as such in this concrete application. Here a Mamdani type fuzzy logic based system is used to approximate unknown nonlinearities associated with the dynamics of the ship steering mechanism. In this method only one adaptive parameter is incorporated, its adaptation is achieved via a Lyapunov

function based rule. On verification with a simulation mode of an ocean-going training ship, the proposed algorithm ensures that the ship steering system is maintained to be asymptotically stable, with the convergence of its tracking error able to converge within a close neighbourhood of the origin. In W. K. Lee et al. [26], a model reference adaptive control design is proposed using output tracking control to synchronise discrete-time chaotic systems. Here the system comprises of two key components the master system for which the respective reference model is based upon and a chaotic slave system, where a Takagi-Sugeno (T-S) fuzzy model is employed to represent its dynamics. Via a gradient descent based algorithm the ideal controller gains are adaptively tuned, such to stabilize the error criterion, even when the two systems begin on varying initial conditions, the slave system under MRAC control approaches synchronisation with the defined reference system. As verification of the proposed adaptive algorithm for application to a T-S fuzzy based discrete-time chaotic system, the synchronisation of two Henon models with different parameters and initial conditions is demonstrated. From this example, the applicability of the proposed method is justified.

In X. Wu et al. [58], a tracking control problem for a nonlinear parameterised system with unknown input nonlinearities is presented. Here nonlinearities with regards to the varying dead-zone and nonlinear characteristics of the control input itself are exhibited. In this scheme, an adaptive feedback controller manipulates the applied actuator featuring control input nonlinearities such to minimise the overall closed-loop tracking control error. The feedback controller incorporates an adaptive back-stepping technique broken into several key steps. In the first step, the initial values and design parameters for the adaptive control laws are initiated, following this the simulation step times are defined and based on the initial states, virtual control inputs and actual control inputs for the state of the system is determined. The new adaptive parameters are then calculated via Lyapunov function based criterion, followed by a newly calculated update of the control input to be sent to the actuator of the control scheme. An advantage of the presented method in comparison to previous studies regarding unknown input nonlinearities is that no prior information with regards to actuator dead-zone characteristics or selection of characteristic design parameters describing input nonlinearities is necessary. Following various simulations based on a classical mass-spring mechanical system, comparisons with a conventional PID control algorithm yield significant reduction in the error between the reference state signal and actual controlled system state on application of the derived adaptive back-stepping controller.

In A. N. Chernodub et al. [59], a neural network based form of MRAC is presented for the adaptive identification and control of a nonlinear dynamic system. Here, a multilayered perceptron (MLP) neural network is used to identify the dynamic system termed by the authors as a neuro-emulator. This neuro-emulator is fed with dynamic inputs, with training of its hidden layer and output neuron weights performed via a gradient decent based algorithm and for comparison an extended Kalman filter training method. The extended feedback neuro-controller is also a multilayered perceptron however featuring one hidden layer, with back-propagation through previously trained neuro-emulator method used to update the neuro-controller neural weights. On simulation with a theoretical nonlinear plant over 100250 samples, the neuro-controller's neural weights were pre-trained during the first 100000 samples, with testing of the controller as a feedback controller with constant neural weights carried out over the remaining 250 samples. From the produced results, four reference model setups with variable time constants were chosen, here it was shown that approximately similar control quality was achieved in terms of the resulting mean square errors (MSE), between the gradient-descent based method of training and extended Kalman filter method. From testing the neuro-emulator with added accuracy criterion, the produced results showed an average mean square error improvement by up to one lower order for the applied extended Kalman filter method. However as seen from this work, a limitation of this case is the rather large iterations of the adaptive algorithm that are necessary to achieve the desired tolerance of mean square error in this MLP based neuro-controller. A further drawback that may also be drawn is in the experimental analysis of the convergence of mean square error, where only a final value is noted and no monitoring is provided in terms of convergence of the square error to its minimum, nor monitoring of convergence of the adaptive neural weights of the neuro-controller during training which would imply the stability of the training algorithm and a measure of robustness of the method, paramount for application to real engineering processes.

In A.S. Kumar et al. [25] a practical application of MRAC in the form of control of a DC motor is presented. In this work, the presented form of MRAC incorporates a fuzzy based adaptive controller used as the main adjustment mechanism of the input signal being sent to the plant, which may also be deemed within the scope of so called direct method of adaptive control. Here, a 3 HP, 240 V 1500 rpm DC motor was tested as a simulated model via Matlab Simulink. From the produced results, a conventional MRAC is compared to that of the presented MRAC with enhancement of a fuzzy controller as the adjustment mechanism, also classified as (MRFAC). From these results the conventional MRAC was not able to

adhere to the desired speed output for lower ranges of rpm as compared to that of the fuzzy controller MRAC. Though on a settled state of the nominal speed, the MRAC achieved less deviation with the desired speed output as compared with the MRFAC, its settling time was proved to be significantly longer with a difference of 1 second. The overall settled output of the MRFAC was within an adequate tolerance of the nominal motor speed, and hence overall justified its applicability towards this concrete engineering application. However, due to the design of the fuzzy adaptive controller IF-THEN rules based on the assumption of good human operator information for the particular system, applicability of this method to a wider sense of engineering processes is limited, furthermore its application to similar processes may not yield as desirable and consistent performance in terms of rate of convergence and adhesion to the desired behaviour of the controller output during adaptation. In C.W. Anderson et al. a model identification based (MIAC) method is used for adaptive control of a simulated heating coil [27]. The results of this study show that although a fixed feedback control, in this case implemented via a PI controller delivers adequate control of the heating coil output air temperature, indeed the combined NN controller in conjunction with a P controller delivered the best results in terms of deviation from the desired set point behaviour of the output air temperature. Here, the authors employ a multilayered NN featuring four inputs, one hidden layer unit and one output layer neuron as the structure of the adaptive controller. It was found that on this system, one hidden layer was adequate for minimising the average square error to 4%. This architecture was concluded to have the best result, amongst individual tests with a standalone PI controller and NN based controller respectively. On analysis of the RMS error of the standalone PI controller compared with the enhancement of the NN based control. The RMS error was reduced further than the PI controller following 500 training epochs. However due to this experimental study being based upon a simulation model of a real heating coil, reliability of the presented form of control to the physical application is dependent on the accuracy of the derived model setup, which as concluded by the authors requires additional experimentation to justify its applicability to control of this heating coil process.

With the design of model reference based adaptive control (MRAC) techniques, stability is key in ensuring convergence of the prescribed adaptive parameters as well as the process output value to its desirable value. From the works [23], [28], [58] the presented MRAC controller designs feature a Lyapunov function based criteria as part of their respective adaptive control laws to ensure stability of convergence of the applied adaptive parameters. Such methods can draw weaknesses with regards to the number of adaptive parameters

necessary to employ the control algorithm, which can increase the complexity and overall computational demand of such form for real time implementation. MRAC design with regards to neural network based architectures and fuzzy logic methods are trending to a wider scope of study and engineering applications. From the works [27] & [59] it can be drawn that although an analysis of square error convergence along with desirable adhesion of the system output to the respective desired set point can outline the success of the applied method, these works lack analysis of the learning process stability for adaptation of their respectively computed adaptive parameters. This being particularly evident in [59], where although convergence of the square error is seen, the addition of monitoring the trajectories of square error convergence and convergence of the identified neural weight parameters may uncover certain instabilities in the employed gradient descent learning algorithm, which are otherwise unnoticed. With the innovation of employing fuzzy based methods, their indeed comes an advantage in conversion of the control parameters from a more quantitatively based approach into that of a more qualitatively based approach, however such methods are highly dependent on human expert knowledge for tuning of the employed rule base. This also suffers the disadvantage of limited applicability and direct application of such methods to similar structured engineering systems. Another point of critique is the lack of experimentation on real engineering processes. The success of most reviewed methods as seen in [25], [27], [57] & [58], [59], are dependent on a well identified or defined models of the applied system, which opens a degree of vulnerability in regards to addressing issues associated with application on a real system, as such unidentified actuator nonlinearities and sensor output disturbances.

2.2 Adaptive Back-stepping Control

A different sense of adaptive control, initially introduced by I. Kanelakopoulos et al. [15] highlights a systematic design approach that progressively stabilises subsystem sets describing the basis of a nonlinear system recursively in order to reach global stability of the nonlinear system as a whole. This method is introduced as a back-stepping form of adaptive control. I. Kanelakopoulos et al. present a nonlinear process as an initial subsystem, where its stability may be described via a known Lyapunov stability function. Following this initial setup, each derived outer shell subsystem acts to stabilise the previous subsystem layer, in a strict-feedback form until the ultimate level of control, where all global stability properties may be established. From this work, various extensions on application of adaptive back-stepping control on a variety of nonlinear systems with strict-feedback structure may be

exhibited. As such, we may recall the work [24] by T.H Liu et al. where a comparison was already drawn in the performance of a presented adaptive back-stepping and adaptive MRAC controller for application on a linear motor system. In the presented control scheme, a strict-feedback scheme of sub-control systems is employed. The drive system stability is thus dictated by uniform continuality of the position and speed error and furthermore, the zero convergence of the position and speed error as the time of simulation approaches towards infinity. From the authors presented results, although the linear motor under loaded conditions with a desired square-wave set was not controlled more adequately with the adaptive back-stepping controller, under comparative tests with a MRAC controller during an unloaded state of the linear motor, the back-stepping controller output was more desirable.

The error measure as an integral of square position error, yielding 0.003801 as compared to the tested MRAC controller with a result of 0.004646. Another example of adaptive back-stepping control may be found in the work [16] by X.P. Liu et al. In their work, a strict-feedback adaptive back-stepping controller design is presented for a general class of stochastic nonlinear systems. Here the state feedback controller is designed regarding constant Markovian switching where stability of the closed-loop is given by its unique solution with an asymptotically stable equilibrium point. In lieu to this, an output feedback controller is derived based on a quartic form of a Lyapunov function such to achieve an asymptotically stable equilibrium point for the closed loop systems unique solution. From the presented simulation result, the control goal was achieved via a convergence of the closed loop system response to zero. Similarly, one may also refer to the work [17] by X. J. Xie et al. a similar architecture of a state-feedback controller design via adaptive back-stepping technique is employed for stabilisation of high-order stochastic systems. The conclusion behind their derived results yields a unique solution with asymptotically stable equilibrium points can be achieved for almost any initial value of the applied system. A drawback from the above mentioned works is that both methods require rather precise description of the applied nonlinear system, either with predefined dynamic models or models with unknown parameters that may be derived through linearity with some known nonlinear functions. This yields a rather large complication to practical application of such control schemes, where certain nonlinear processes are unable to be precisely described.

An enhancement to this and a trending area of research for adaptive back-stepping techniques is the collaboration with neural network and fuzzy logic based methods, particularly for control of nonlinear systems with presence of unknown non-linear functions.

A recent work to mention is the application of backstepping design via higher order neural networks (HONNs) [20] (Figure 3). In their work, a feedforward neuro-backstepping control is used for control of an unknown MIMO plant. For efficient real-time training the extended Kalman filter (EKF) is used to train the neural weights of a single HONN approximator, which is used for approximating the desired virtual controls and ideal practical controls of a block strict feedback form (BSFF) with incorporated backstepping technique, as in real application the precise system model and accurate description of additional disturbances is rather trivial to compose. The stability of the update rule and further control scheme are further verified via Lyapunov control function analysis.

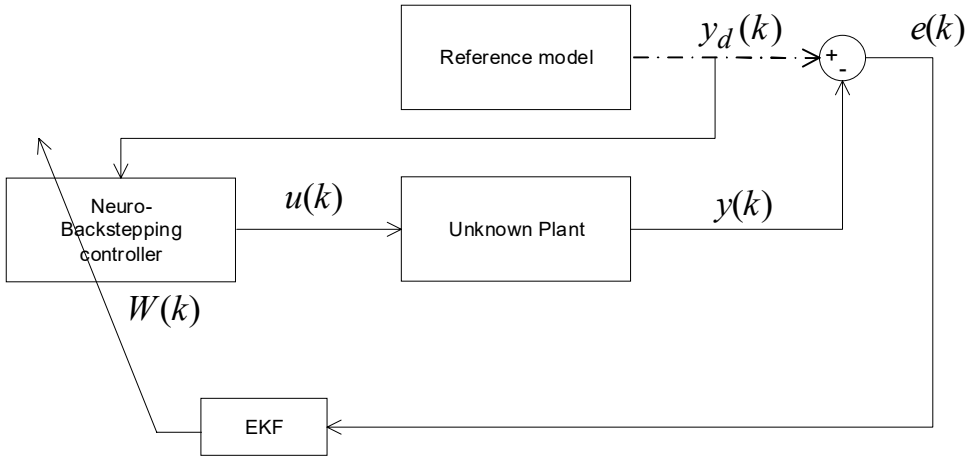


Figure 3- Discrete-time HONN Backstepping Control Scheme as presented in [20]

B. Chen et al. [18] present an adaptive fuzzy tracking controller via back-stepping technique capable of controlling a class of nonlinear SISO systems with uncertain nonlinearities to within a close neighbourhood of the proposed desired signal. Here a Mamdani based fuzzy logic system is used to directly approximate the desired and unknown control signal for the nonlinear SISO process, contrary to methods as such the works of S.C. Tong et al. [60], where unknown nonlinear functions and constants are used as a component of the control law, which also shows the increased dimensionality in the amount of adaptive parameters within the control scheme. A virtual control signal as a result is constructed which is used as a parameter within the adaptive law for update of the real tracking control law of input *u.* Simulation on a third order theoretical nonlinear system, yields that in comparison with an implemented adaptive controller by Zhang, Ge et al. [61], both the adaptive neural and adaptive fuzzy controllers deliver close adhesion of the desired output response. However, the adaptive fuzzy controller exampled achieves quicker adaptation with only one run of the adaptation rule, versus 15 runs of the neural controller rule, deeming it to be computationally efficient. Furthermore, connotations to replacement of the fuzzy based control signal

approximator with that of a neural network based form are noted as a possible enhancement to the presented methodology.

Similar to the adaptive back-stepping controller structure presented in B. Chen et al. [18], Z. Liu et al. [62] present an adaptive fuzzy output feedback based controller. Here too, a fuzzy based system is used to approximate the unknown control signals as well as desired signals and an adaptive back-stepping technique is used to design the adaptive controller. The proposed design guarantees all signals of the closed loop system to be semi-globally bounded and the control error converges to a close neighbourhood around a zero centred origin. The difference in this method is the higher number of unknown adaptable parameters present within the algorithm as opposed to one in the sense of [18]. On demonstration with a third-order nonlinear system with un-modelled dynamics and dynamical disturbances, the performance index of mean square error was evaluated for individual solutions to the system of first order equations representing the third-order nonlinear system. On two tested sets of chosen constant design parameters the largest mean square error was 0.1372, implying close adhesion to the desired response of the controlled nonlinear system. A recent work by S. Heidari et al. [19] presents another example of a fuzzy based system incorporated with back-stepping technique as an adaptive control design. Here a fuzzy (Proportional-derivative) PD based system is used to compensate system model uncertainties with a back-stepping controller, in conjunction with Lyapunov based adaptive laws used to control a continuum robot manipulator. In addition to this, the implemented back-stepping controller incorporates an additional fuzzy logic based system to increase the robustness of the back-stepping controller. The conclusions drawn from this study are that under an applied step response with an online updateable gain, the fuzzy based back-stepping controller achieves 0.48 seconds rise time versus 0.6 seconds with the standalone back-stepping controller technique. Further to this the proposed method shows better adhesion to the desired behaviour of the robot manipulator under applied disturbance, as well as quick response time in achieving the desired set point value, verifying suitability of the applied method. With regards to nonlinear system control, adaptive back-stepping design and its application has proven itself applicable to wide sense of nonlinear dynamic systems with particular successes in processes with unknown nonlinear functions and input nonlinearities.

In works [16], [24], the successful application of adaptive back-steeping design for zero convergence of the respective nonlinear system state variables was presented, with application to high-order stochastic nonlinear systems exampled in [17]. A drawback from

these methods is the necessity of rather precise description of existing nonlinear properties which is difficult to provide in certain practical applications. Thus, adaptive fuzzy or neural network based techniques have been used in conjunction with adaptive back-stepping design, to model the unknown system nonlinearities due to their proven approximation capabilities. Another consideration as exampled in [19] & [48] is the use of fuzzy logic for direct approximation of the desired control input signal as opposed to approximation of unknown system nonlinearities as in [60]. However, as exampled from [62] the necessity to estimate a higher number of adaptive parameters in the employed algorithm indeed leads to higher computational expense. These methods further neglect actuator faults which often can occur on real system application and is thus a further negative with a majority of the reviewed adaptive back-stepping design methods.

2.3 Optimal Control and Dynamic Programming Design for SISO Processes

O. Kovalenko et al. [63] (Figure 4) present a practical application of adaptive control in the sense of an automotive fuel-injection system. Here the authors derive an adaptive critic design (ACD) which principled upon approximate optimal control over time. As an initial, a neural network (NN) based form of the process model was derived. Here time-lagged recurrent neural networks (TLRNs) were used as a basis for the investigated engine modelling. The dynamic modelling of the engine intake manifold features 4 input neurons, 7 hidden layer neurons and 2 output neurons with recurrent connections. Tests associated with application to the manifold pressure and mass air flow rate as the system output, yields quite successful dynamic modelling which serves as a good platform for an extension to control. For this engineering process the main outputs for control lay in control of the engine torque and air-fuel ratio. In this closed loop control setup, the adaptive critic controller design utilises an error function (difference between desired and measured output) as a mechanism for calculation of a local cost function. A sum of this local cost function (U) and discount factor γ (collectively J which is minimised over time) is used to deliver the critic network output Q as an approximation of J. The optimal controller is thus obtained by training an action network using the output signal which is provided by the critic network output. In this study, the authors use a NN structured controller of 4 input neurons, 7 hidden layer neurons and 3 output neurons. After 15 cycles of training or epochs, adequate control was achieved in terms of both the engine torque control and air-fuel ratio control with a minimum tracking error of 26 units after 15 cycles of training.

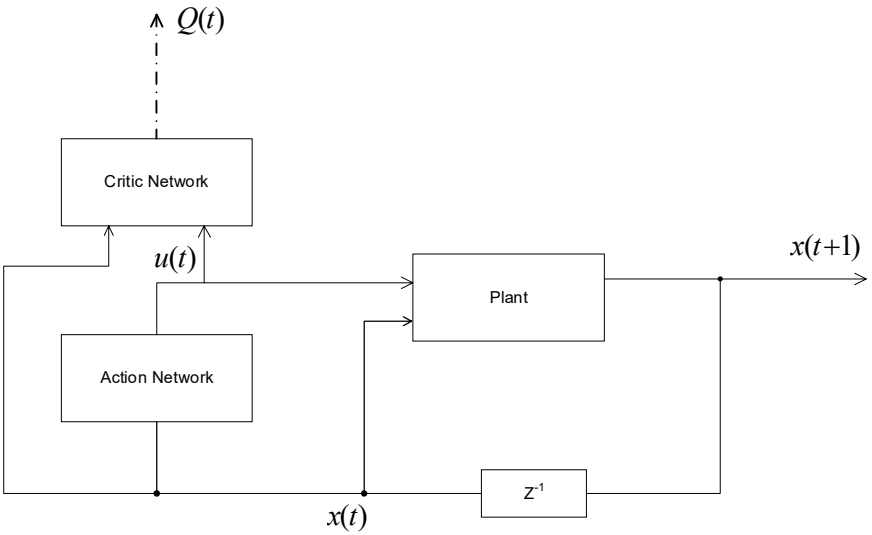


Figure 4- A typical scheme of an action-dependent heuristic dynamic programming (ADHDP)as presented in [63]

In a more recent study by D. Liu et al. [30], a dynamic programming technique for optimal control of unknown nonlinear systems is derived. In this presented control scheme, the authors propose a three-layer feed-forward neural network based architecture, divided into three consecutive blocks. The first element in this control scheme is the model network which once trained via the gradient descent based adaptation rule serves as the identified model of the nonlinear system. An action network with similar gradient descent based adaptation techniques is used to control the model network, influenced by the third element of the control scheme being the critic network. Here the role of the critic network is to adaptively achieve convergence of the cost function to the optimal cost function $J*$ with optimal control law $u*$, via an adaptive dynamic programming based technique, parameters of the ADP algorithm itself, further adapted via gradient descent based training of the critic network neural weights. The newly calculated control law in each iteration is used as a key parameter in the gradient descent based update rule for the weight adaptation of the action neural network. This method may further be deemed as an indirect parameter update mechanism for adjusting the controller parameters used for manipulation onto the applied system. Simulation results on application to a theoretical nonlinear process yield after identification of the model network via 1000 time increments using 100 samples of training data, the critic and action networks are trained in 120 iterations over 2000 training steps within each iteration, for desirable accuracy. The results of this setup yield rapid convergence of the cost function and its derivative to the optimal cost function value along with convergence of the respective state trajectories to the origin, validating its feasibility for control of discrete time systems with unknown nonlinear dynamics.

A further example of adaptive dynamic programming technique may be found in D. Liu's most recent works [31]. Here a three-layer feed-forward architecture is used, among two neural network blocks. Contrary to the works [63] & [30], the intermediate model network is omitted, with direct application of the action network to the plant being employed. The critic network block in this proposed scheme utilises a gradient descent based weight update rule, to adaptively tune the output function of the critic network. An enhancement in this method is the derived policy iteration algorithm proposed by the authors, which is shown to reduce computational expense of the critic network component within the control scheme. The action network is thus updated further via a similar gradient descent based technique, incorporating the critic networks output as a part of its weight update rule. Tests on both linear and nonlinear systems show successful convergence of the monitored performance index function to optimal values within a short period of applied iterations. Further to this, tests on a concrete physical problem of a torsional pendulum system yield quite desirable results. There, both the critic and action neural networks are trained for 400 samples followed by iterative control over 100 further samples. The system states are stabilised within the first 40 samples and convergence of the performance index function to its optimal value is achieved within 30 iterations, justifying its suitability for stabilisation of such exampled nonlinear systems and real-time control. On the other hand one may drawn several points for discussion. In this work neural networks are used as a key mechanism behind the employed control scheme. With this indeed comes the element of approximation error of the critic network performance index function and iterative control law function approximated by the action network where their exact values cannot be achieved. A further remark is in the convergence property of the iterative performance index function and stability analysis of the system during stages of the iterative control law, both of which lack justification in the results of this work.

In S. Formentin et al. [64], a comparison is drawn between "model-based" and "data-based" approaches for adaptive control. Here an indirect model based method is analysed where the process model itself is derived based on measured data from the engineering process. The applied controller parameters may be calculated as an extension of the previously determined model, where the controller output is determined solely from the measured input and output data of the engineering process. In this paper, the author's main contributions are with regards to the data-based approach via a derived optimal controller in lieu with a correlation based approach. The tested process consists of a three mass torsionally flexible shaft, driven by a servo motor. The authors present a resulting measure of variance between the measured

output and reference model of 1.0915×10^{-3} for a model based tuned controller and 0.8486×10^{-3} in the sense of the data based approach. The results of this test along with that derived on a theoretical discrete plant model is that the data-based approach achieves the least measure of variance between the measured output and reference model. With the conclusion that if the model-based approach features a model structure of substantially higher order, as in the cases studied, the data based approach may statistically outperform the model-based form in terms of control effort.

As another area of computational intelligence techniques for adaptive control design, dynamic programming particularly used as a tool for optimal control problems indeed exhibits strong potentials for nonlinear system control problems. With this said, one may also note certain limitations regarding its implementation. One drawback that can be noted is in the structure of the employed adaptive algorithms. This can often concern the so called curse of dimensionality, convergence towards the optimal solution of which being in cases rather computationally demanding. Further to this, an area necessary for deeper study that may be recalled from the works [63] & [30] is the convergence and stability properties of the employed adaptive dynamic programming techniques. This particularly raises an issue in justifying the robustness and overall stability of the control scheme during real-time online adaptive control of similar natured nonlinear dynamic systems and hence is an opening for further study.

2.4 Closed Loop Feedback Methods of Adaptive Control for SISO Processes, via Polynomial Function Based Control

Another field in adaptive control which has achieved substantial growth in recent times is polynomial function based neural networks (PNNs) for adaptive identification and control. An early example of the use of PNNs for dynamic system identification may be recalled in the work [35] by G. P. Liu et. al. In this work an on-line identification scheme featuring PNNs is used for nonlinear system identification. This method utilises previously defined polynomial based functions comprising of previous inputs and outputs to the dynamic system, as a part of vector ϕ along with its product with updatable neural weights vector W and residual error ε to determine to real-time approximation of the dynamic nonlinear system output. In order to update the neural vector weights, a recursive learning algorithm is employed as a modification of the least square method learning algorithm, with accompanying Lyapunov based convergence criterion for monitoring properties of the adapted neural weights. On simulation of this method to theoretical nonlinear systems, even

exampled with a nonlinear system output contaminated by white noise, the authors conclude successful identification within limits of the desired error tolerances, indicating the suitability of this method for online dynamic system identification.

Since this work, a more recent development to remark is M. Gupta et al. [65]. This publication sets a modern overview behind various platforms and architectures of neural networks for both adaptive identification and control of dynamic systems. Works from M. Gupta (2002-2003) present higher-order-neural-networks (HONNs) comprised of higher-order-neural-units (HONUs) that are able to capture not only linear correlation between components of the applied system input patterns but also higher-order correlation between the components of the input patterns. Further to this, from works as early 1993 (Taylor & Commbes) [66] HONNs are proven to be computationally efficient, with excellent properties in pattern recognition as well as being realisable on a hardware basis. These HONNs presented as a class of PNNs feature updatable neural weights, trained via supervised learning algorithms like the incremental gradient decent (GD) or Recursive Least Squares (RLS) method or in the sense of batch training the back propagation through time algorithm [67] (BPTT) realised via the Levenberg-Marquardt equation termed in this dissertation work in short as the L-M algorithm. Following the works of M. Gupta et al. [65] various publications have highlighted the growing potentials of HONUs for dynamic system adaptive identification and control. Following the publication I. Bukovsky and N. Homma [34], the terminology of HONU evolved from the initial works of M.Gupta. The term Linear Neural Unit (LNU) was introduced to denote first order higher order neural units which by architecture are analogical to the well known ADALINE (adaptive linear element) as a single layered neural network. For second order polynomial architectures, the term Quadratic Neural Unit (QNU) was introduced, with further orders termed as the cubic (CNU) and quartic neural unit for third and fourth order architectures respectfully.

In I. Bukovsky and N. Homma [34], a quadratic neural unit (QNU) is analysed as a powerful medium for plant modelling and control. The authors summarise a classification of the QNU (as a class of HONUs) into several key categories. Here such neural units having only previously measured data of the real system being used as its input vector are noted as static QNUs with those featuring previous neural model outputs as part of their input vector, being termed as dynamic QNU or (DQNUs). Their implementations may be further broken into a discrete time or continuous time form. Following various experimental observations, the authors conclude several key properties regarding the performance of the QNU. This work

concludes that QNU architectures have smaller variance of neural outputs when trained from various initial conditions as compared to MLP NNs. Furthermore, that HONUs have adequate performance regarding nonlinear approximation, where during their training the problem of convergence to various local minima is absent contrary to other forms of conventional neural networks. Connotations are drawn to the promising performance of higher orders of neural network architectures, with regards to further reduction in sum of square errors over a similar training period of the neural models respective neural weights.

The extension of this result is seen in a more recent study by M. Gupta et al. [36]. Here the theories presented in [65] regarding dynamic modelling with such HONNs is reinforced. The HONU of second order as a second-order-neural-unit (SONU), further termed as the quadratic neural unit (I. Bukovsky and N. Homma [34]) undergoes deeper theoretical analysis in terms of its pattern recognition and learning performance abilities compared with that of a static MLP neural network. In this case study, a multiple input static function is trained via the various architectures for system identification. What is seen from their results is that the HONU is able to reach convergence to a minimum sum of square errors (SSE) in just a few training epochs of the applied Levenberg-Marquardt algorithm. A comparison with the tested MLP with 10 neurons in its hidden layer and a linear output neuron achieves a similar order of its SSE in 3500 applied epochs of the applied training algorithm, highlighting the computational efficiency of the investigated HONU. Furthermore, the higher orders of the applied HONU that were employed deemed to have more rapid reduction in its SSE over the same training data and training method, highlighting its superiority in computational efficiency to calculate a more precise approximation of the applied training data.

I. Bukovsky et al. [33] investigate the potentials of HONUs for fast state feedback control of unknown nonlinear systems. Here the authors present the structural schematics of such form of control. There as an initial, a HONU is used to model the dynamics of the investigated system. Following adequate system identification another HONU is used in state feedback form to manipulate the newly feed inputs of the investigated process. Their results show that on investigation of a theoretical nonlinear plant, the unknown nonlinear plant under pulse wise desired behaviour behaved with rather large oscillatory response to the fed system inputs. However, the applied cubic-neural-unit (CNU) in state feedback with online tuned neural parameters, still manages to adhere considerably well to the overall desired system response. On further investigation with the presented nonlinear plant, now of variable

parameters in terms of a variable damping factor, the plant undergoes instability after 150 seconds of simulation time. In spite of this, the applied CNU is still able to not only stabilise, but further control the unstable nonlinear plant with adequate adhesion to the overall desired system behaviour, highlighting the robustness of the controller. A further notion drawn from this study is the application of such control scheme with HONU architecture to a linear plant. The authors conclude that the higher order neural units, namely QNU and CNU deliver faster performance in terms of desired convergence under a step response, as well as reduced overshooting implying the suitability of such HONU architectures for control loop optimisation purposes.

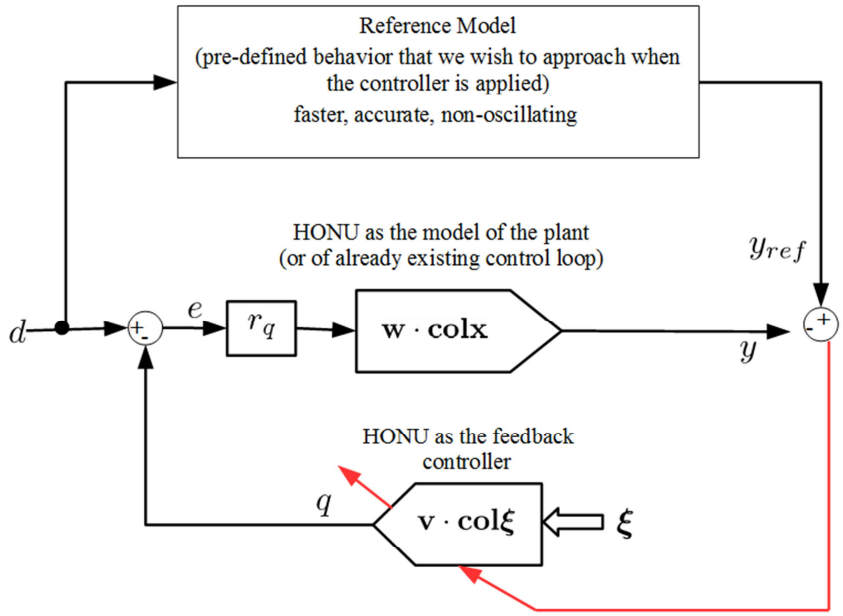


Figure 5- Higher order neural unit model reference based control scheme (HONU-MRAC), where one HONU is identified as a plant and one standalone HONU controller is extended in feedback (possibility of multiple HONU controllers in feedback also presented in [68]).

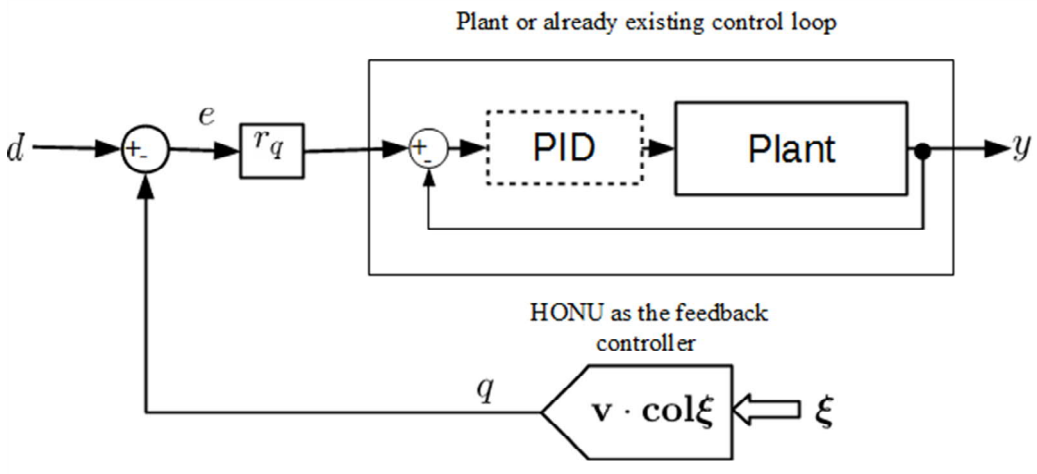


Figure 6- Higher order neural unit model reference based control (HONU-MRAC) for existing control loop optimisation, as presented in [32].

Following the above mentioned works on HONUs, a most up to date overview of the applied theories of HONUs from the fundamental architectures and supervised learning algorithms through to implementation may be found in (Figure 5 & Figure 6). In terms of recent real-time adaptive control applications a modern example of HONU adaptive control on a SISO process may be found in the work [32] by L. Smetana. In this work the presented setup incorporates the use of the QNU (I. Bukovsky and N. Homma [34]) for both adaptive identification of a Bathyscaphe system in CTU, as well as an adaptive real-time state feedback controller for process optimisation. Here an offline tuned dynamic quadratic neural unit (DQNU) was trained via the GD algorithm, for adequate identification of the full dynamics of this process whilst achieving reasonable reduction in mean square error over 1500 epochs or runs of the training algorithm. Following offline tuning of the DQNU representing the system model, an online QNU as a state feedback controller (Neuro-controller) is applied. Here the state feedback controller is adapted online via the GD algorithm, using the outputs of the DQNU system and deviation of the output from the applied reference model and real system output, denoted shortly as e_{reg}. On real time application on the Bathyscaphe system with pulse wise desired behaviour, the Neuro-controller displays significant improvement in adhering to the desired setpoint as compared to a conventional P controller. A comparative test of switching on the Neuro-controller after 150 seconds of only a P controlled plant, shows the remarkable improvement in reduction of the control error e_{reg} peaking with a deviation of 40 mm only on the first several samples following a change in the desired pulse signal, as compared to 60 mm previously. The monitoring of the neural weights of the adaptive controller (Neuro-controller) in real time also justifies the stability of the employed GD training algorithm with all weights of the Neuro-controller settling to a steady value within 400 seconds of application on the system. Following comparison of the Neuro-controller and its performance with a standalone P-controlled Bathyscaphe system, another test was conducted using an applied PID controller tuned adequately within the working range of the experimental setup. From the authors experimental results, it was shown that the PID controller indeed improved the behaviour of the Bathyscaphe system under the applied desired setpoints. However, in the initial samples of a step wise change in the desired setpoint value, the PID controller proved to have large degrees of overshooting, more so pronounced for a larger range of applied desired steps. On a small scale range of applied steps the concluding result of this work was shown, where the Neuro-controller applied to an adequately tuned PID controller delivered significant performance in improved adhesion to the applied desired behaviour of the Bathyscaphe

process. Here it was concluded that the use of a nonlinear QNU controller with real time adaptive control algorithm via GD as extended to the Bathyscaphe system deemed to not only have faster response time towards the settled desired behaviour of the system, but also the property of reduced overshooting. This real time result is also a justification of the exhibited properties of such nonlinear HONU controllers as mentioned in the works [33], [34]. A more recent work featuring extended applications to a water levitation and two-tank liquid level system control may be found in Section 5.1, which further highlights the advantages of HONU feedback control algorithms as constant parameter controllers implemented in real-time.

HONUs as class of PNN based methods, have proven themselves to be an adequate medium for initial control of engineering systems with unknown nonlinearities or nonlinear systems by natural structure, as well as a suitable tool for controller optimisation as applied to linear SISO processes. From the works of Gupta et. al and I. Bukovsky [65] & [34], we may draw a vast comparison between the use of conventional NNs as such that of MLP architecture and the use of HONUs. A drawback that may be ascertained from using MLP architectures as a means of system modelling or extension to control is their necessity to use rather complex training algorithms in lieu with longer training runs to achieve adequate convergence to a minimal square error. In contrast to this, the approximation strength of such conventional NNs may be improved by adding more neurons or even additional layers, which can provide better approximation for nonlinear systems. However as can be seen in the work [33] (I.Bukovsky, S.Redlapalli) for efficient real time learning algorithms as such that of GD and L-M batch training algorithms, HONUs are computationally faster in achieving adequate convergence in square error whilst achieving desirable control performance for both nonlinear unknown systems as well as linear systems of SISO structure. One may yield a drawback of applying higher orders of HONUs is the dimensionality issue that arises in processing such large weight update matrix particularly in the sense of real time adaptive control algorithms, further for systems which require rather high sampling incorporated into their respectively identified adaptive models. As a rather novel concept in MRAC control, though HONUs have shown promising results as exhibited in [32] for real-time process control. Another challenge which still exists in this field is the guarantee of stability of the proposed nonlinear dynamical model or nonlinear control loop. Therefore, as a further motive of this dissertation work the proceeding subsection aims to overview key works associated with stability analysis of NN based adaptive control.

2.5 Control Loop Stability Methods for Neural Network based Adaptive Control

Another topic which remains nowadays is in the guarantee of stability of the proposed adaptive model or adaptive control loop. In a recent review of recurrent neural network based methods for stability [48] a natural conclusion yields that we have no universally best stability evaluation algorithm for NN as for nonlinear dynamical systems, and that proper stability evaluation is always case dependent, for different aspects of the control loop design. With regards to neural network (NN) forms of adaptive control numerous works have been published with concern to stability of convergence during learning of a polynomial function based approximator. Most recently [39] where the fundamental architecture and learning algorithms of HONUs are overviewed accompanied by a spectral radius based learning criteria for monitoring stability during the applied adaptive update law for identification of the respective HONU model weights. However, though the approach guarantees stability in real-time during application of the adaptive algorithm, it does not prove the stability of the HONU model during application on new process data, which for certain engineering processes may be advantageous for reapplication of existing controller parameters or continued operation due to change of mechanical system parameters. Thus, a further area which has received particular attention in recent times, is the assessment of global asymptotic stability (GAS) of the adaptive control loop as a whole, with further intention led towards minimising the amount of online computations during application of the controller algorithm. Earlier methods for global stability analysis of neural networks have been focussed on linearization analysis and Lyapunov function based techniques for a class of Hopfield neural networks [69] & [70]. Some recent examples include the use of homeomorphism and Lyapunov function based techniques by [71] where global robust exponential stability of the equilibrium point for delayed neural networks is developed. In [72] problems concerning stability for static neural networks with time delays are investigated, via a delay partitioning technique a novel stability criteria justified via the Lyapunov-Krasovskii criteria is investigated to warrant the global stability of the investigated static NNs. An earlier method necessary to remark maybe found in the work [73] and his works there in. There, the assessment of GAS of the respective equilibrium points for all non-decreasing neural network activation functions is presented for a class of dynamic layered neural networks. Should the neural model be defined in state matrix form, the negative of the positive diagonal matrix T serving as an interconnection matrix with respect to a nonlinear diagonal mapping matrix g, if proven to be quasi-diagonally column-sum dominant the authors proof yields existence and uniqueness of such equilibrium point along with justification of global

asymptotic stability. With focus to recurrent neural networks (RNNs) methods, an example advantageous for nonlinear systems with bounded nonlinearities may be found in [41]. However, till this date one of the most common and readily employed approaches for assessing asymptotic stability of recurrent neural networks, is that of the Linear Matrix Inequality (LMI), an example of such method maybe found in [42] and further [43] where the Lyapunov-Krasovskii stability theory along with the linear matrix inequality approach are used to determine stability of both constant and time-varying delayed neural networks.

In practical control applications however, although the stability of a system equilibrium point may be justified, it is often necessary and more practical to ensure that the process states are bounded with respect to bounded process inputs that are supplied to the engineering process. As a result, studies focussed towards bounded-input-bounded-output (BIBO) and further bounded-input-bounded-state (BIBS) stability has been readily published. In [74] a practical condition is established termed as input-output stability which addresses that if the engineering process input satisfies a given amplitude constraint, under what condition will the process output satisfy the given amplitude constraint. Furthermore, under what condition will the process output satisfy certain energy constraints for the given energy constrained inputs, their results show a practical stability test method focussed on linear time-varying systems. A more modern example of input-output form of stability may be found in [75] where stability analysis of Markovian jump systems with time-varying delay is presented, due to an innovation of transforming the conventional Markovian jump system with additional time-varying delay to an input-output form, the stochastic stability of the closed loop system may be guaranteed. Another consideration is with regards to stability of process states given constrained process inputs, leading to the definition of input-to-state stability (ISS). In [45] the fundamental concepts of input-to-state stability are defined which states a condition for the process state vector in a given time point to be constrained less than a sum of a respectively defined κL and κ_∞ class functions for the system, leading to an effective stability analysis approach which holds for any input $u(\cdot)$ given any initial condition x^0 over all time point greater the zero. Further in his works, an extension of this method may be found in [76] and for a more recent example of extended application of the bare principles of ISS, in [77] where Lyapunov functions are employed to construct the input-to-state stable control laws for design of model predictive control (MPC) aimed for discrete-time systems. With regards to the conversion of continuous time frame ISS definition to that of a discrete time frame, a clear platform may be found in the work [78]. Several recent examples with focus to recurrent neural networks include [46] for ISS verification of a class of time-delayed

RNNs and in [79] where the Lyapunov-Krasovskii functional method and LMI techniques are used for ensuring exponentially stable ISS on multiple time-varying delay RNNs. Backboned from ISS, a more precise definition is bounded-input-bounded-state stability (BIBS). The definition of BIBS provides a more specific definition which relates the evaluation of a process state transition matrix to define functions encompassing the properties of a κL function which is in essence asymptotically stable through time and a κ_∞ function which is always increasing as time is greater than zero. A simplification and sufficient condition for BIBS for definition of bounded states with respect to bounded process inputs about a given process region may be found in [44].

Following comprehensive review of state-of-the-art adaptive control approaches and further a review of suitable stability analysis methods, this dissertation as outlined in the key objectives (section 3) aims to build on the existing theories of HONU adaptive identification and control, being one such trending platform for real industrial practice of nonconventional control. To do so, the proceeding section recalls the latest theories in HONU architectures, fundamental learning algorithms and its application to feedback control (HONU-MRAC).

3 Objectives and Contributions of this Dissertation Work

The scope of this dissertation is aimed at advancing the field of adaptive identification and closed loop control via HONUs. As an initial component, deeper study into the design and practical application of HONU-MRAC control for industrial systems is provided. Then, a key contribution of this dissertation is the extension of a new framework for stability assessment of HONUs and their respective closed loop stability. Following a comprehensive review behind state-of-the art methods the following objectives of this dissertation may be defined:

1) *Recall and extend on theories of HONU adaptive control*: Following review of state-of-the-art approaches, this dissertation as a first objective recalls the latest theories of HONUs and their fundamental learning algorithms in chapter 4. A further contribution of this dissertation is to detail the derivation of an adaptive feedback gain parameter and provide deeper experimental analysis and comparison of the Recursive Least Squares (RLS) algorithm for HONU-MRAC control.
2) *Propose new pointwise state-space representation of a HONU*: Recalling the fundamentals of HONUs and more classical theories of nonlinear discrete time

system modelling (further methods as via LOLIMOT technique and NQR (or SDRE approach) [2], [3]), this dissertation as a major contribution derives the transformation of nonlinear HONU architectures to a linearly approximated state-space model in discrete state points. Following this newly derived state-space form the Discrete-time HONU Stability Condition (DHS) is developed as a local pointwise state based evaluation of BIBO stability and further proven for justifying asymptotic stability in neighbourhood of the evaluated state point via Lyapunov stability theory.

3) *Propose a new pointwise state-space representation of a HONU via polynomial decomposition*: Due to an intrinsic relation of HONU architectures to discrete time nonlinear state space models via their in-parameter factorization, another major objective and contribution of this dissertation is to derive a new decomposition approach for modelling HONU models and further their whole HONU-MRAC control loop. Due to this intrinsic relation as extension of works [89]-[90] a decomposition of the HONU polynomial equation to sub-polynomial state space form is proposed.
4) *Derive a new ISS based stability condition for BIBS stability assessment of HONU polynomial architectures*: From the results of 3), the concepts of ISS stability are extended to the derived decomposed state space form and a method for justification of BIBS stability, termed as Decomposed Discrete-time HONU Stability Condition (DDHS) and further DDHS(Strict) condition is developed. An added advantage of this decomposition approach is the preservation of dynamical accuracy for higher order nonlinear polynomial structures (presented in this dissertation for up to 3rd order as a HONU-MRAC control loop).
5) *Experimental analysis to validate the proposed DHS and DDHS approaches*: A final objective and contribution of this dissertation work is to provide deep experimental analysis and comparison of the derived DHS and DDHS approaches, where the DHS method further serves as one means of validation of the DDHS approach. The methods are also analysed with respect to Lyapunov control function stability approach. All methods as an ultimate contribution of this dissertation work are implemented on physical industrial systems with focus to rail automation applications to validate the feasibility of use for our modern industry. Future directions via sliding mode control approach with the presented HONU decomposition and multi-layered HONU feedback control for low to moderately nonlinear systems are also discussed.

4 Theoretical Background of Adaptive Identification and Control with HONUs

In this section the most recent achievements for SISO dynamic system identification [100]-[102] and control [80], [87], [91], [103]-[105] via HONUs are recalled. From these works, HONUs have proven to be a very effective tool for both linear and low to moderately nonlinear system identification and MRAC based adaptive control. Throughout this dissertation, the term "low to moderately nonlinear" refers to dynamic systems that can be approximated by a given order of polynomial nonlinearity i.e. r. With the application of HONUs, it is assumed that the applied dynamic system is stable either naturally, or via an existing stabilization control loop in order to obtain valid training data for identification.

As a majority of industrial process applications fall within the class of linear or low to moderately nonlinear dynamic systems, quite often linear or lower order HONUs are sufficient for fast adaptation to real-time changes, with the application of linear neural units (LNU) also applicable to weakly nonlinear systems with sample-by-sample adaptive identification to learn temporal non-linear system dynamics. In general, selection of the appropriate polynomial order is case dependent, where higher orders are chosen only in case that linear or lower order neural units cannot met the requirements of fast response to a change in system inputs, or cannot sufficiently identify non-linear or oscillatory dynamics of a given engineering process. Therefore, this dissertation focusses on 1st to 3rd order HONUs i.e. linear neural units LNU (HONU, $r=1$), quadratic neural units (HONU, $r=2$) and cubic neural units (HONU, $r=3$). As may be recalled from the works [33] the classical notation of a HONU is the summation of multiplication terms that is, for example for a QNU or CNU i.e. HONU for $r=2$ and $r=3$, represented as follows

$$\tilde{y}=\sum_{i=0}^{n_x}\sum_{j=i}^{n_x}\mathbf{W}_{i,j}x_i x_j,\quad \tilde{y}=\sum_{i=0}^{n_x}\sum_{j=i}^{n_x}\sum_{\kappa=j}^{n_x}\mathbf{W}_{i,j,\kappa}x_i x_j x_\kappa, \tag{1}$$

where $x_0=1$ is the augmented unit (neural bias) of the input vector, $\mathbf{W}$ denotes a corresponding multidimensional array of the HONU. The classical notations (1) recently published in and most currently [39], proposed in a long-vector representations as

$$\tilde{y}=\mathbf{w}\cdot col^{r=2}(\mathbf{x}),\quad \tilde{y}=\mathbf{w}\cdot col^{r=3}(\mathbf{x}), \tag{2}$$

where $\mathbf{w}$ represents the long-vector of all neural weights. Therefore, an r dimensional weight matrix $\mathbf{W}$ flattens e.g. for a QNU to

$$\mathbf{w}=\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{i,j} & \dots & w_{n_x,n_x} \end{bmatrix}=\begin{bmatrix} \{w_{i,j}\} & i=0\dots n_x \\ & j=i\dots n_x \end{bmatrix}. \tag{3}$$

Further, an augmented input vector of neural inputs $\mathbf{x}$ yields a long-column vector form $col^{r=i}(\mathbf{x})$. In the sense of a QNU its respective elements may be calculated as follows

$$col^{r=2}(\mathbf{x})=\{x_i x_j; i=0..n_x, j=i..n_x\}. \tag{4}$$

Analogically, the elements corresponding to a CNU may be calculated as follows

$$col^{r=3}(\mathbf{x})=\{x_i x_j x_\kappa; i=0..n_x, j=i..n_x, \kappa=j...n_x\}. \tag{5}$$

The weight-update system of such HONU models is generally given then via following relation

$$\mathbf{w}(k)=\mathbf{w}(k-1)+\Delta\mathbf{w}; \Delta\mathbf{w}=\Delta\mathbf{w}(\mathbf{w}(k-1)). \tag{6}$$

In terms of the update rule of respective neural weights, Table 1 shows details of three fundamental supervised learning algorithms, where $\mathbf{J}$ is the Jacobian matrix and $\mathbf{I}$ denotes the identity matrix. Further, the variable e denotes the neural error that may be defined as follows

$$e(k)=y(k)-\tilde{y}(k). \tag{7}$$

On extension of a HONU feedback controller, similarly to equations (1) & (2) the output value computed by the HONU feedback controller is given by a summation of multiplication terms thus for a QNU (i.e. HONU for $\gamma=2$), the HONU controller output may be denoted as follows

$$q=\sum_{i=0}^{n_\xi}\sum_{j=i}^{n_\xi}\mathbf{V}_{i,j}\xi_i\xi_j=\mathbf{v}\cdot col^{\gamma=2}(\boldsymbol{\xi}), \tag{8}$$

or for a CNU (i.e. HONU for $r=3$) we may denote the HONU controller output as

$$q=\sum_{i=0}^{n_\xi}\sum_{j=i}^{n_\xi}\sum_{\kappa=j}^{n_\xi}\mathbf{V}_{i,j,\kappa}\xi_i\xi_j\xi_\kappa=\mathbf{v}\cdot col^{\gamma=3}(\boldsymbol{\xi}), \tag{9}$$

where $\mathbf{V}$ represents the $\gamma-$dimensional array of neural weights corresponding to the HONU feedback controller, $\mathbf{v}$ is its corresponding long-vector representation and $\mathbf{v}.col^{\gamma=i}(\boldsymbol{\xi})$ is the representative long-vector multiplication form. Here the variable $col^{\gamma=i}(\boldsymbol{\xi})$ now comprises of the variables ξ_i, ξ_j in the sense of a QNU or further ξ_κ in the sense of a CNU. The elements of which follow analogically from relations (3)-(4), comprising of step-delayed process outputs y or HONU model outputs $\tilde{y}$ further, the desired behaviour of the process d. For simplification throughout further text, the index of the polynomial order $r=i$ and further

the use of parenthesis in the operators will be dropped. Therefore, the result of $col^{r=i}(\mathbf{x})$ and $col^{\gamma=i}(\boldsymbol{\xi})$ will be respectively denoted as **colx** or **col**ξ to symbolise the respective HONU input vector. With regards to the new inputs u that are fed back to the identified HONU model, the following control law holds for a QNU feedback controller

$$u = d - p \cdot \sum_{i=0}^{n_\xi} \sum_{j=i}^{n_\xi} \mathbf{V}_{i,j} \xi_i \xi_j = d - p \cdot \mathbf{v} \cdot col^{\gamma=2}(\boldsymbol{\xi}), \tag{10}$$

In the sense of a CNU controller, the control law can be written as

$$u = d - p \cdot \sum_{i=0}^{n_\xi} \sum_{j=i}^{n_\xi} \sum_{\kappa=j}^{n_\xi} v_{i,j,\kappa} \xi_i \xi_j \xi_\kappa = d - p \cdot \mathbf{v} \cdot col^{\gamma=3}(\xi), \tag{11}$$

where p denotes a feedback gain and d denoting the plant controllers desired set-point. As presented in earlier works [87]-[88] for certain engineering applications it may be more advantageous to due to scale of the static plant gain being of much higher order in comparison to the supplied plant inputs, to apply p as a compensation at the control input as opposed to in feedback, in such case the following control law yields e.g. for a QNU

$$u = p \cdot \left(d - \sum_{i=0}^{n_\xi} \sum_{j=i}^{n_\xi} \mathrm{V}_{i,j} \xi_i \xi_j \right) = p \cdot \left(d - p \cdot \mathrm{v} \cdot col^{\gamma=2}(\xi) \right), \tag{12}$$

where for a CNU, the control law may be expressed as

$$u = p \cdot \left(d - \sum_{i=0}^{n_\xi} \sum_{j=i}^{n_\xi} \sum_{\kappa=j}^{n_\xi} v_{i,j,\kappa} \xi_i \xi_j \xi_\kappa \right) = p \cdot \left(d - p \cdot \mathrm{v} \cdot col^{\gamma=3}(\xi) \right), \tag{13}$$

For a whole block scheme, refer to Figure 5.

TABLE 1

SUMMARY OF FUNDAMENTAL WEIGHT UPDATE LAWS FOR HONUS OF GENERAL ORDER

Incremental Training Algorithms	
Gradient Descent (GD)	$\Delta \mathbf{w} = \mu \cdot e(k) \cdot \mathbf{colx}^T$
Normalised Gradient Descent (NGD)	$\Delta \mathbf{w} = \dfrac{\mu}{\left\|\mathbf{colx} \cdot \right\|_2^2 + 1} \cdot e(k) \cdot \mathbf{colx}^T$

Recursive Least Squares (RLS)	$\mathbf{R}(0)=\frac{1}{\delta}\cdot\mathbf{I}$ $\mathbf{R}(k)^{-1}=\frac{1}{\mu}\cdot(\mathbf{R}^{-1}(k-1)$ $-\frac{\mathbf{R}^{-1}(k-1)\cdot\mathbf{colx}(k)\cdot\mathbf{colx}(k)^T\cdot\mathbf{R}^{-1}(k-1)}{\mu+\mathbf{colx}(k)^T\cdot\mathbf{R}^{-1}(k-1)\cdot\mathbf{colx}(k)})$ $\Delta\mathbf{w}=\mathbf{R}^{-1}(k)^{-1}\cdot\mathbf{colx}(k)\cdot e(k)$
Batch Training Algorithms	
Levenberg-Marquardt (L-M)	$\Delta\mathbf{w}=(\mathbf{J}^T\cdot\mathbf{J}+\frac{1}{\mu}\cdot\mathbf{I})^{-1}\cdot\mathbf{J}^T\cdot\mathbf{e}$ $\mathbf{J}(k,:)=\mathbf{colx}^T$
Resilient Backpropagation	$\Delta\mathbf{w}=\Delta\mathbf{w}\left(\Delta\left(\nabla\mathbf{E}\right)\right)$
Conjugate Gradients	$\Delta\mathbf{w}=\alpha\left(\mathbf{r_e},\mathbf{p},\mathbf{J}\right)\cdot\mathbf{p}\left(\beta(\mathbf{r_e},\mathbf{J})\right)$

Regarding the parameter update rule of the HONU controller, the neural weight update rule of the HONU feedback controller can be analogically derived as follows

$$\Delta\mathbf{v}=-\frac{\mu}{2}\cdot\frac{\partial {e_{ref}}^2(k)}{\partial\mathbf{v}}=\mu\cdot e_{ref}(k)\cdot\frac{\partial[y_{ref}(k)-\tilde{y}(k)]}{\partial\mathbf{v}}, \tag{14}$$

where the reference error is defined in (20), thus

$$\Delta\mathbf{v}=\text{-}\mu\cdot e_{ref}(k)\cdot\frac{\partial\tilde{y}(k)}{\partial v_i}. \tag{15}$$

Considering equations (1)-(2) it generally results for a HONU feedback controller of arbitrary polynomial order that

$$\Delta\mathbf{v}=\mu\cdot e_{ref}(k)\cdot\mathbf{w}\cdot\frac{\partial\mathbf{colx}}{\partial\mathbf{v}}, \tag{16}$$

where for the example of QNU the long vector of partial derivatives yields to be as follows

$$\Delta\mathbf{v}=\mu\cdot e_{ref}(k)\cdot\mathbf{w}\cdot\frac{\partial\mathbf{colx}^{r=2}}{\partial\mathbf{v}}, \tag{17}$$

$$\frac{\partial\mathbf{colx}^{r=2}}{\partial\mathbf{v}}=\left\{\frac{\partial x_i}{\partial\mathbf{v}}x_j+x_i\frac{\partial x_j}{\partial\mathbf{v}};i=0..n_x,j=i..n_x\right\}. \tag{18}$$

Analogically for a CNU (i.e. for $r=3$)

$$\frac{\partial\mathbf{colx}^{r=3}}{\partial\mathbf{v}}=\left[\left\{\begin{matrix}\frac{\partial x_i}{\partial\mathbf{v}}x_jx_\kappa+x_i\frac{\partial x_j}{\partial\mathbf{v}}x_\kappa+x_ix_j\frac{\partial x_\kappa}{\partial\mathbf{v}};\\ i=0..n_x,\ \ j=i..n_x,\ \ \kappa=j..n_x\end{matrix}\right\}\right]. \tag{19}$$

However, $\Delta\mathbf{v}$ may be updated over every batch of the experimental data. In this case, the L-M algorithm according to Table 1 may be employed. Here e_{ref} denotes neural error that may be defined as follows

$$e_{ref}(k) = y_{ref}(k) - y(k) \cong y_{ref}(k) - \tilde{y}(k). \tag{20}$$

On establishing the HONU feedback controller weight update mechanism, it is clear that whether employing a sample-by-sample update rule or batch form of training the overall weight update rule yields to be

$$\mathbf{v}(k) = \mathbf{v}(k\text{-}1) + \Delta\mathbf{v}; \Delta\mathbf{v} = \Delta\mathbf{v}(\mathbf{v}(k\text{-}1)), \tag{21}$$

where the parameter $\mathbf{v}$ represents the HONU feedback controllers neural weights, the tuning of which may be performed via either online or offline training as per the algorithms denoted in Table 1. In certain engineering processes it may be advantageous to employ an adaptive proportional gain of the HONU feedback controller e.g. in processes where the static gain at different working ranges may vary causing increased steady state deviation. As an enhancement to the recalled applied theory behind HONU architectures, this dissertation additionally derives an adaptive feedback gain p in the control law relations (10)-(11), published in the work [95]. From the principles of the GD algorithm we may define the fundamental update rule of parameter p as follows

$$\Delta p = -\frac{1}{2} \cdot \mu_p \cdot \frac{\partial e_{ref}{}^2(k)}{\partial p}, \tag{22}$$

where μ_p denotes a unique learning rate to that of the derived HONU feedback controller or further, identified HONU process model. According to the definition (20) it yields that

$$\Delta p = -\frac{1}{2} \cdot \mu_p \cdot e_{ref}(k) \cdot \frac{\partial [y_{ref}(k) - \tilde{y}(k)]}{\partial p}. \tag{23}$$

However, the partial derivative of the reference model output with respect to the adaptive proportional gain p is zero, thus from (1) the respective HONU model output is in fact a conjunction of the neural weight vector $\mathbf{w}$ and the neural input vector $\mathbf{colx}$, therefore the incremental update law for the adaptive feedback gain p generally results for an arbitrary polynomial order as

$$\Delta p = \mu_p \cdot e_{ref}(k) \cdot \mathbf{w} \cdot \frac{\partial \mathbf{colx}}{\partial p}. \tag{24}$$

Where in the example of a QNU, the resulting partial derivatives yield to be as follows

$$\Delta p = \mu_p \cdot e_{ref}(k) \cdot \mathbf{w} \cdot \frac{\partial \mathbf{colx}^{r=2}}{\partial p}, \tag{25}$$

$$\frac{\partial \mathbf{colx}^{r=2}}{\partial p} = \left\{ \frac{\partial x_i}{\partial p} x_j + x_i \frac{\partial x_j}{\partial p}; i = 0..n_x, j = i..n_x \right\}. \tag{26}$$

Further, for a CNU the resulting partial derivatives are thus extended as follows

$$\frac{\partial \mathbf{colx}^{r=3}}{\partial p} = \left\{ \frac{\partial x_i}{\partial p} x_j x_k + x_i \frac{\partial x_j}{\partial p} x_k + x_i x_j \frac{\partial x_k}{\partial p}; i = 0..n_x, j = i..n_x, k = j..n_x \right\}. \tag{27}$$

From this section the fundamental architectures of HONUs for adaptive identification and MRAC based control were recalled. With addition of the rather novel application of the RLS learning algorithm for application to HONU adaptive identification and control and further the derivation of an adaptive proportional gain p for application to a HONU feedback controller. In the proceeding section a practical case study of a two-tank liquid level system with several configurations are presented in a bid to compare the various combinations of applied learning algorithms in terms of their performance in rate of convergence. Further, several real rail automation examples are analysed to highlight the advantages and disadvantages of the overviewed HONU-MRAC design.

5 HONU-MRAC Control Loop Design for Real Industrial Applications

Following an overview of the key HONU architectures and respective learning algorithms for real time adaptive identification and control, this section aims to study their application to real industrial process control. As an initial a two-tank liquid system is studied, with focus on investigating the various learning algorithms as in table 1. Then two comprehensive studies of real rail automation systems are presented to highlight the applicability for low to moderately nonlinear system control. A further main objective in chapter 8 is then presented to analyse stability of the derived HONU-MRAC control loop designs via the proposed DHS and DDHS approaches.

5.1 *Two-tank Liquid Level System*

Given the applied theories of HONU process identification and control as presented in section 4, let us consider a Two-Tank Liquid Level system illustrated in Figure 7.

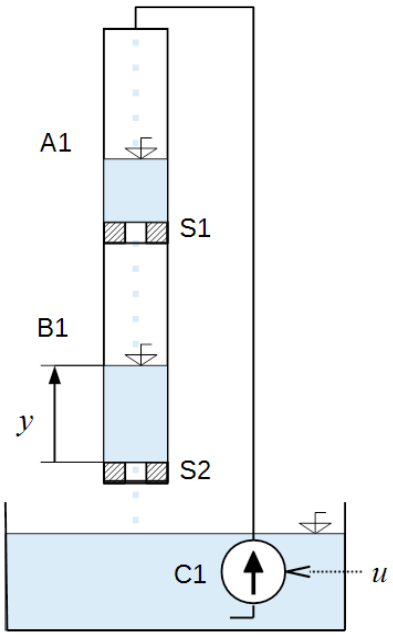


Figure 7 – Schematic overview of the Two-Tank Liquid Level system.

The process dynamics of the upper tank may thus be described as follows

$$A \cdot \frac{dh_1}{dt} = Q_t - C_{db} \cdot s_1 \cdot \sqrt{2 \cdot g \cdot (h_1 - h_2)}. \tag{28}$$

Here, $Q_t[m^3 s^{-1}]$ denotes the inlet flow rate of the tank system. C_{db} represents an orifice discharge coefficient, A denotes the cross-sectional area of the upper and lower tanks. s_1 denotes the cross-sectional area of the first tank exit orifice, with h_1 and h_2 resembling the height of the water level for the tank 1 and 2 respectively. Then it follows that we may similarly represent the dynamics of the second tank

$$A \cdot \frac{dh_2}{dt} = C_{db} \cdot s_1 \cdot \sqrt{2 \cdot g \cdot (h_1 - h_2)} - C_{dc} \cdot s_2 \cdot \sqrt{2 \cdot g \cdot h_2}, \tag{29}$$

where s_2 resembles the cross-sectional area of the lower tank orifice and C_{dc} is the discharge coefficient of the lower tank system. Considering that the orifices are ideal, we may see that an induced nonlinearity is present in the system via the square root law in the flow balance equations (28)-(29). For the proceeding study the following parameters are considered. The tank cross-sectional area $A = 0.002[m^2]$, orifice cross-sectional areas $s_1 = s_2 = 0.000785[m^2]$, orifice discharge coefficients $C_{db} = C_{dc} = 0.60$, the density of water $\rho = 1000[kg/m^3]$ and gravitational constant of acceleration $g = 9.81[ms^{-2}]$. It follows from

initial testing as per Figure 8, that the system features varying degree of static gain about different operating points of the tank system.

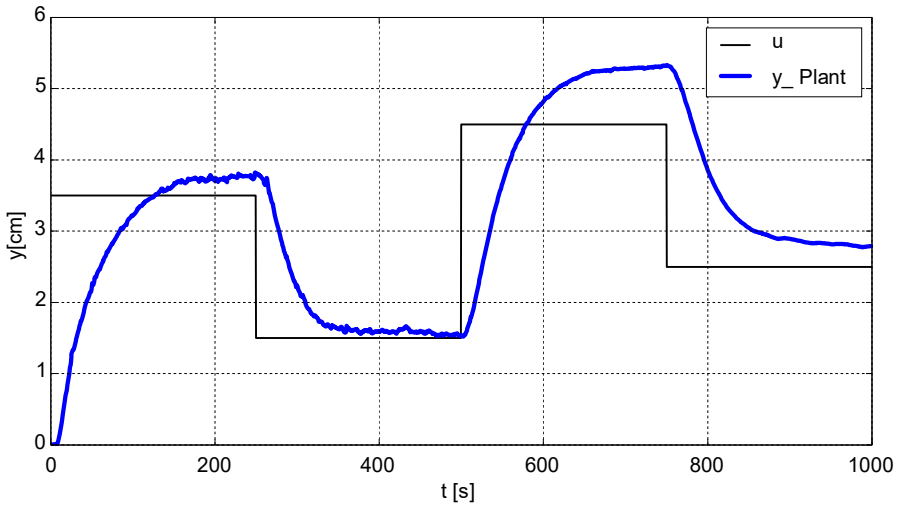


Figure 8-Standalone operation of real nonlinear two-tank water level system with varying degrees of static gain.

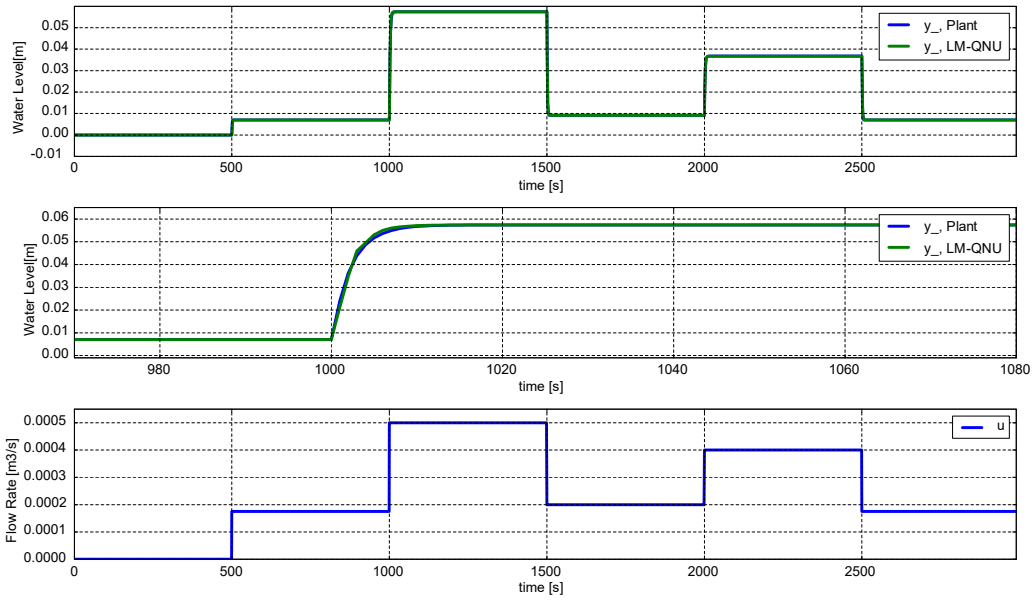


Figure 9 - HONU plant identification of QNU trained via L-M supervised learning algorithm for $n_y = 4$ and $n_u = 3$ applied to Two-Tank Liquid Level system. As presented in [91] and [92].

For levels of input pump voltage u, ranging between 1–5 [V] the systems natural output response yields higher deviations of static gain on level greater than 2.5 [cm] as opposed to lower water levels in the range of 1–2.5 [cm]. Following simulation of the mathematical relations depicted in (28)-(29), several key architectures of HONU neural units were tested. Following numerous tests, it was found that for adaptive identification of the Two-Tank Liquid Level system experimental data, a QNU with L-M form of neural weight training deemed to be the most suitable configuration to successfully identify the dynamics of the

tank system. Here 4 previous neural unit outputs along with 3 previous system inputs were used for identification as depicted per Figure 9.

TABLE 2
HONU-MRAC CONTROL LOOP PARAMETER CONFIGURATIONS

HONU Plant Model	HONU Controller Training	No. HONU Model Outputs	No. of Desired Value Inputs	Learning Rate (without normalization)	Epochs
QNU_LM	GD	4	3	0.001	50
QNU_LM	LM	4	3	0.001	50
QNU_LM	RLS	4	3	0.001	10
QNU_RLS	RLS	4	3	0.115	5

Following this table 2 summarises the parameter configurations for the various learning algorithms used on a QNU (HONU, $\gamma = 2$), all HONUs are identified as standalone adaptive feedback controllers as part of the HONU-MRAC closed control loop.

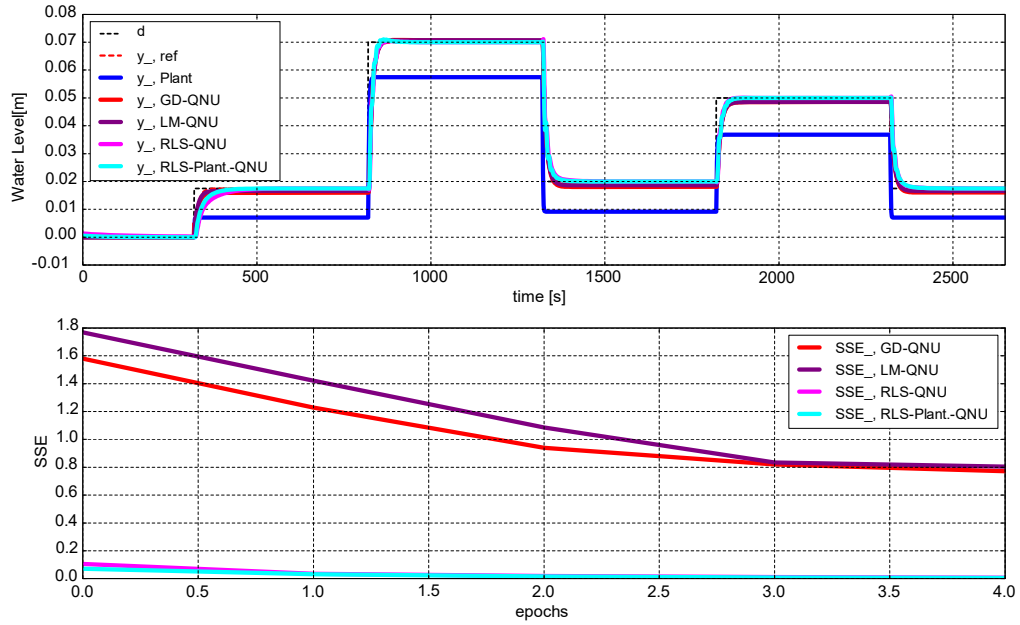


Figure 10 – HONU-MRAC closed control loop performance for control of the lower tank water level h_2 of the Two-Tank Liquid Level system. As presented in [91] and [92] , results generated from [98].

Figure 10 and Figure 11 illustrate the performance of the various QNU (HONU, $\gamma = 2$), adaptive controllers identified as per table 2. From these results, we may note that all forms of learning algorithms were able to deliver desirable control performance in terms of adhesion to the respective setpoints as well as reduction in sum of square errors (SSE). From Figure 10, it is intriguing that a slightly lower steady state margin is achieved following results of the RLS training algorithm in much fewer epochs as compared to the GD and L-M training algorithms, where only 5 epochs as opposed to the GD and L-M algorithm with a

respective 50 epochs of neural weight training were necessary to achieve a similar order of SSE.

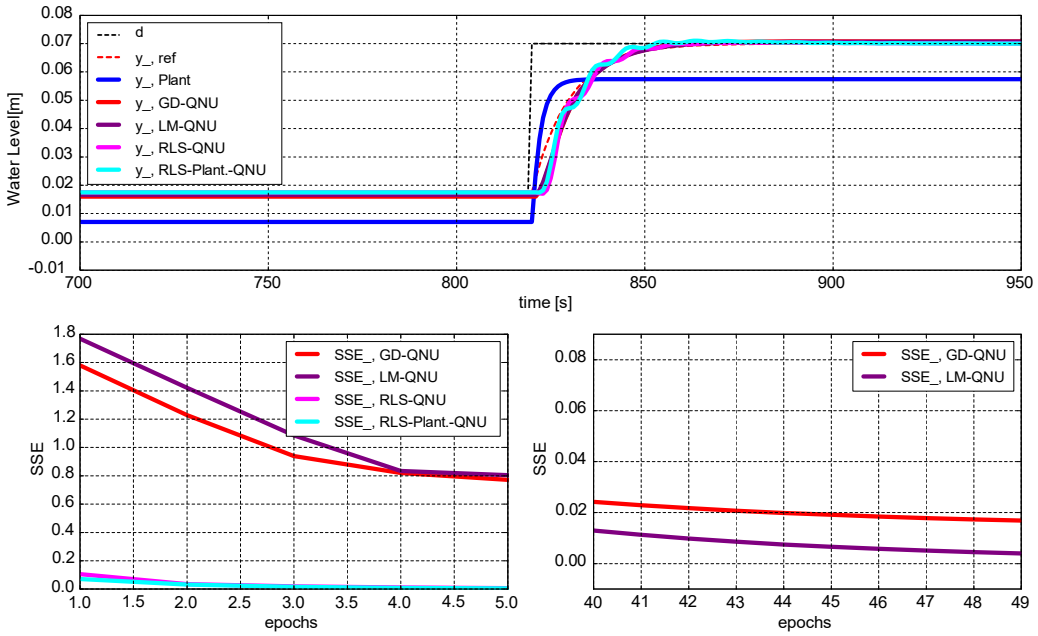


Figure 11 – Detail of HONU-MRAC control loop performance in convergence of SSE between different supervised learning algorithms for Two-Tank Liquid system control.GD and L-M algorithm reduced to a similar order of SSE after 50 epochs. As presented in [91], [92], [97] & [107] , results generated from [98].

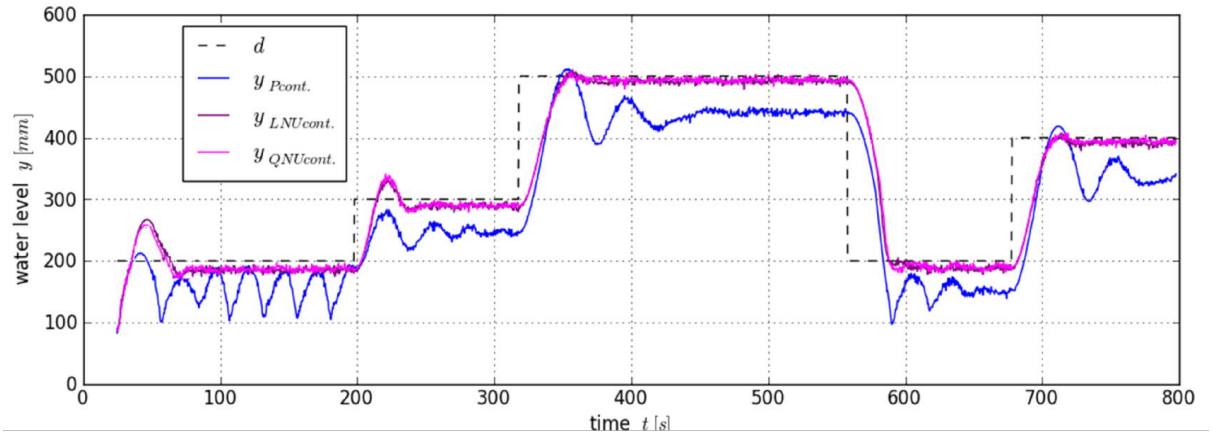


Figure 12 - Real-time HONU-MRAC control loop of constant parameters, trained with L-M for process identification followed by RLS pre-training with further L-M training algorithm as a feedback controller for the Two-Tank Liquid Level system with an existing P-control loop. As presented in [91], [92], [97] & [107].

A further interesting notion from this result is the speed of convergence via application of the RLS method for adaptive identification of the plant followed by adaptive control of an applied HONU as a feedback controller. Here Figure 11 illustrates in detail the slightly more optimal performance delivered by application of the RLS algorithm in both the adaptive identification and control phase which is a promising direction for future study, especially in the sense of real-time adaptive identification and control applications. Following analysis of the results depicted in Figure 10 and Figure 11, a QNU (HONU, $\gamma = 2$) feedback controller

with a L-M training algorithm for process identification, followed by a RLS pre-training of the HONU feedback controller in lieu with 10 further training epochs via the L-M batch training algorithm yields superior performance in controller optimisation on extension to a P-control loop as depicted in Figure 12. From these results, all analysed supervised learning algorithms serve as fast, computationally efficient learning algorithms for process control of linear and further nonlinear engineering systems. From the Two-Tank Liquid Level system application, the RLS algorithm in lieu with extended training via the L-M algorithm shows promising potentials for fast, real-time adaptive control of practical engineering processes, and hence is a further investigated in this dissertation work.

5.2 *Experimental Railway Stand (CTU Roller Rig) Active Wheelset Control*

Another intriguing example in this dissertation work is the design of HONU-MRAC control for the CTU experimental railway stand (CTU Roller Rig) situated in the Czech Technical University in Prague. A current problem under investigation is the control of lateral skew of an independently rotating wheelset (IRW), primarily to maintain an adequate margin of deviation between the Roller Rig wheel flanges and rail heads in a bid to minimise the wheel and rail wear along with reduction of the horizontal skewing forces. As the scope of this work is in the design and stability analysis of a HONU–MRAC control loop for SISO engineering processes, this section will be organised to first analysis the design and dynamical behaviour of the CTU Roller Rig, followed by a detailed design for real-time dynamic identification and control of its lateral skew problem. Since its initial conception from the late 80's the CTU Roller Rig has been an important tool for investigation of control engineering problems with regards to active wheelset guidance of scaled railway wheelset mechanisms. In its current state the CTU Roller Rig features two independently actuated wheelsets situated on a bogie frame.

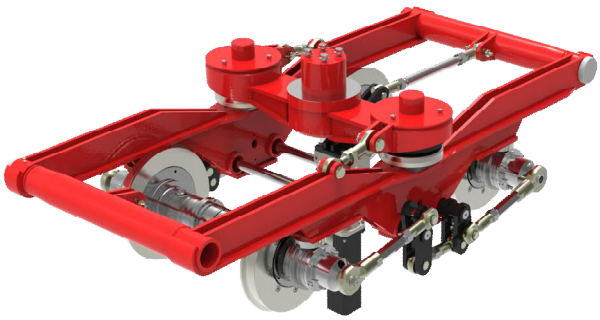


Figure 13 - Illustration of the CTU Roller Rig (2014). Featuring configurable 2 IRW or conventional wheelsets with central servo motor drive for active yaw torque steering (Figure courtesy of the Dpt. of Automotive, Combustion Engine and Railway Engineering, FME CTU).

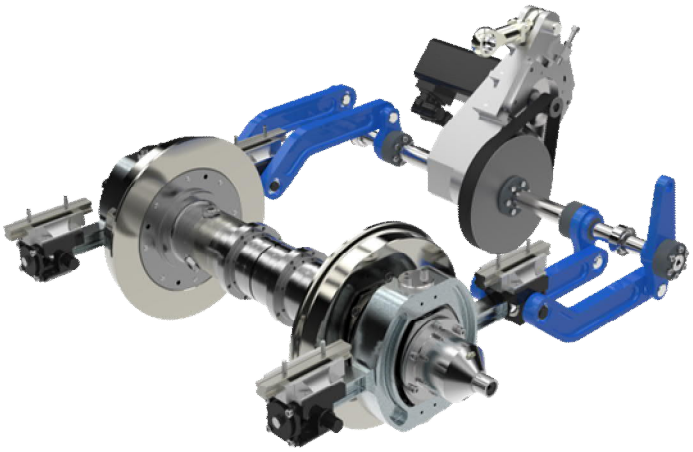


Figure 14-State-of-the-art CTU Roller Rig actuated wheelset steering mechanism (2015-2016) [81]–[83]. See control results in Figure 24-Figure 26 and also stability analysis results in Figure 53 & Figure 54 (Figure courtesy of the Dpt. of Automotive, Combustion Engine and Railway Engineering, FME CTU).

The respective design is a 1 to 3.5 scaled version of a real railway vehicle design. It features a wheelbase 714 mm a track gauge of 410 mm and individual wheel diameters of 263 mm. Each wheelset sits on a set of 700 mm diameter rollers driven via 2 asynchronous AC motors to simulate the behaviour of the Roller Rig under both straight as well as curved track profiles. Figure 13 illustrates a recent setup of the CTU Roller Rig dated to 2015, as depicted in this figure, the front and rear wheelsets are manipulated via a central servo drive which via mechanically coupled linkages are able to guide the respective wheelset in the required direction. From this conception in latter 2015-2016, a redesign of the existing rig was performed to improve behaviour under active wheelset guidance. Currently each respective wheelset may be configured to either a conventional setup or for operation as an IRW. In this design, illustrated in Figure 14, the CTU Roller Rig may be actuated via 2 separate servo drives for each wheelset (Multiple-input-single-output (MISO)), with a new belt pulley drive to control the respective Roller Rig wheelset via active yaw torque control to the coupled steering linkages, or via a Single-input-single-output (SISO) configuration considering only the leading wheelset, for actuation and measured lateral skew with respect to the central span of the whole bogie frame. For the scope of this dissertation the Roller Rig control task is minimised to a SISO configuration. The control objective is to ensure the lateral position of the leading wheelset from the central axis given either a conventional wheelset or IRW configuration. Furthermore, this dissertation will focus on control for the speed range of 50-80 rpm.

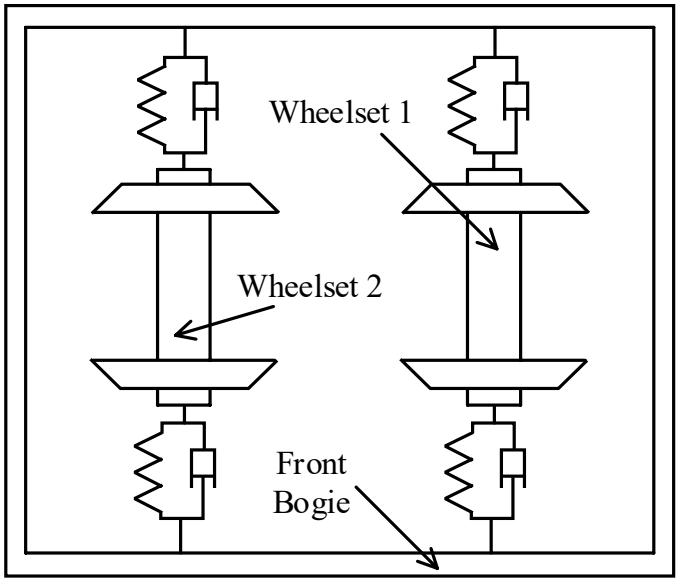


Figure 15-Simplified schematic of a railway vehicle bogie frame with two wheelsets for theoretical analysis in (30)-(35).

To understand the dynamical behaviour of the CTU Roller Rig as a standalone mechanism without the means of lateral skew control, Figure 15 illustrates a simplified mathematical model of a railway bogie with two wheelsets. Given the above schematic, we may introduce the following model parameters adopted from the real roller rig system as per Table 3.

TABLE 3
PARAMETERS OF THE CTU ROLLER RIG

Variable	**Value**	**Description**
m	62.8 [kg]	Mass of an individual wheelset
m_b	150 [kg]	Mass of the bogie frame
f_{11}	1.35x105 [N]	Longitudinal creep coefficient (= 0 for simple IRW wheelset)
f_{12}	1.21x105 [N]	Lateral creep coefficient
I_w	2.9 [kgm^2]	Wheelset yaw inertia
I_b	9.5 [kgm^2]	Bogie yaw inertia
l	0.214 [m]	Half gauge
l_b	0.5 [m]	Half gauge between wheelsets
K_w	4000[kN/m]	Primary lateral stiffness per wheelset
C_w	120[kN s/m]	Primary lateral damping per wheelset
λ	0.1	Conicity
r	0.1315 [m]	Wheelset radius
V	2.77 [m/s]	Speed

Therefore, given the above parameterisation, we may proceed to construct the necessary equations of motion of the above simplification of the roller rig setup, with the assumption of neglecting the frictional forces in the bearing housings and bearing box of each wheelset bogie, along with the small mechanical clearances in the steering mechanism.

The Leading Wheelset (Wheelset 1)

$$m\ddot{y}_{w1} = \left[\frac{-2f_{11}}{V} - Cw\right]\dot{y}_{w1} - Kw.y_{w1} + 2f_{22}\Psi_{w1} + Cw.\dot{y}_{b1} + Kw.y_{b1} + Cw.l_b\dot{\Psi}_{b1} + Kw.l_b\Psi_{b1}, \tag{30}$$

$$I_w\ddot{\Psi}_{w1} = \frac{-2f_{11}l\lambda}{r}y_{w1} - \frac{2f_{11}l^2}{V}\dot{\Psi}_{w1} + T_{\Psi 1}. \tag{31}$$

The Trailing Wheelset (Wheelset 2)

$$m\ddot{y}_{w2} = \left[\frac{-2f_{11}}{V} - Cw\right]\dot{y}_{w2} - Kw.y_{w2} + 2f_{22}\Psi_{w2} + Cw.\dot{y}_{b1} + Kw.y_{b1} - Cw.l_b\dot{\Psi}_{b1} - Kw.l_b\Psi_{b1}, \tag{32}$$

$$I_w\ddot{\Psi}_{w2} = \frac{-2f_{11}l\lambda}{r}y_{w2} - \frac{2f_{11}l^2}{V}\dot{\Psi}_{w2} + T_{\Psi 2}. \tag{33}$$

The Bogie Frame

$$m_b\ddot{y}_{b1} = Cw.\dot{y}_{w1} + Kw.y_{w1} + Cw\dot{y}_{w2} + Kw.y_{w2} - 2Cw.\dot{y}_{b1} - 2Kw.y_{b1}, \tag{34}$$

$$I_b\ddot{\Psi}_{b1} = Cwl_b\dot{y}_{w1} + Kw.l_by_{w1} - Cw.l_b\dot{y}_{w2} - Kw.l_by_{w2} - 2Cw.l_b{}^2\dot{\Psi}_{b1} - 2Kw.l_b{}^2\Psi_{b1}. \tag{35}$$

From relation (30)-(35) it follows that we may further compose the respective state-space representation of the CTU Roller Rig system, in the following form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx, \end{aligned} \tag{36}$$

where A corresponds to the matrix of dynamics of the CTU Roller Rig system, B denotes the matrix of input coefficients with u representing the control input values, which are in fact the independent yaw torques for the leading and trailing wheelset respectively. Further, the state variables vector denoted shortly as x.

$$A=\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{Kw}{m} & \frac{1}{m}\left[\frac{-2f11}{V}-Cw\right] & \frac{2f22}{m} & 0 & 0 & 0 & 0 & 0 & \frac{Kw}{m} & \frac{Cw}{m} & \frac{Kw.l_b}{m} & \frac{Cw.l_b}{m} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{2.f11.l.\lambda}{Iw.r} & 0 & 0 & \frac{2.f11.l^2}{Iw.V} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{Kw}{m} & \frac{1}{m}\left[\frac{-2f11}{V}-Cw\right] & \frac{2f22}{m} & 0 & \frac{Kw}{m} & \frac{Cw}{m} & \frac{Kw.l_b}{m} & \frac{Cw.l_b}{m} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{2.f11.l.\lambda}{Iw.r} & 0 & 0 & \frac{2.f11.l^2}{Iw.V} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\frac{Kw}{m_b} & \frac{Cw}{m_b} & 0 & 0 & \frac{Kw}{m_b} & \frac{Cw}{m_b} & 0 & 0 & \frac{-2Kw}{m_b} & \frac{-2Cw}{m_b} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\frac{Kw.l_b}{I_b} & \frac{Cw.l_b}{I_b} & 0 & 0 & \frac{-Kw.l_b}{I_b} & \frac{-Cw.l_b}{I_b} & 0 & 0 & 0 & 0 & \frac{-2l_b^2Kw}{I_b} & \frac{-2l_b^2Cw}{I_b}
\end{bmatrix} \tag{37}$$

$$B=\begin{bmatrix}
0 & 0 & 0 & \frac{1}{Iw} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{Iw} & 0 & 0 & 0 & 0
\end{bmatrix}^T \tag{38}$$

$$u=[T_{\Psi 1} \quad T_{\Psi 2}]^T \tag{39}$$

$$x=\begin{bmatrix} y_{w1} & \dot{y}_{w1} & \Psi_{w1} & \dot{\Psi}_{w1} & y_{w2} & \dot{y}_{w2} & \Psi_{w2} & \dot{\Psi}_{w2} & y_{b1} & \dot{y}_{b1} & \Psi_{b1} & \dot{\Psi}_{b1} \end{bmatrix} \tag{40}$$

$$C=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{41}$$

With relations (37)-(41), we may analyse the behaviour of our model in the Matlab Simulink environment to understand the behaviour of the roller rig wheelsets lateral skew due to an applied yaw torque as the input to each wheelset. For the scope of the work, as the focus of the lateral skew control design will be aimed for a SISO configuration, the below simulation results will be illustrated for manipulation of only the leading wheelset (wheelset 1). As can be seen from the initial simulation under conventional wheelset behaviour, the lateral skew of the leading railway wheelset undergoes rather undamped oscillations (Figure 16), whilst the IRW shows a clear veering towards one track direction with regards to its lateral skew (Figure 17).

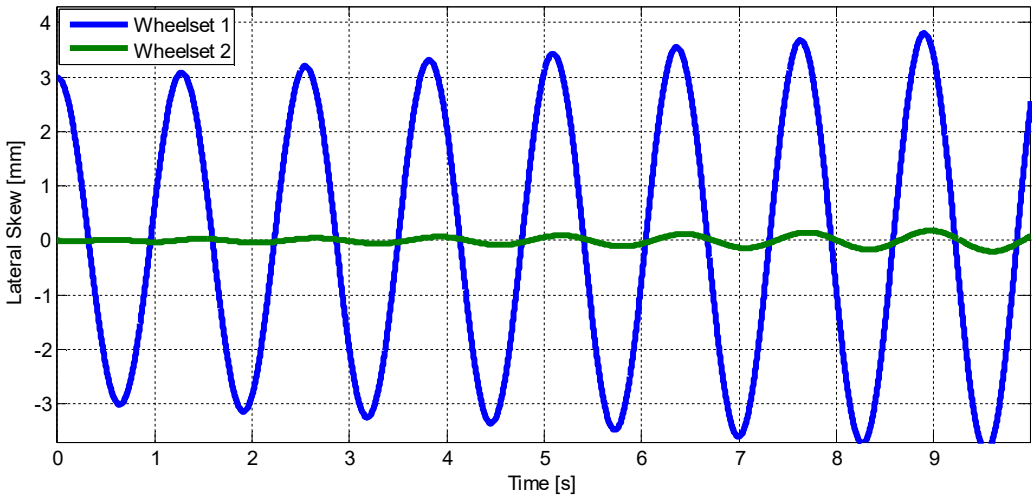


Figure 16-Simulation of the roller rig mathematical model (37)-(41) for a conventional wheelset configuration. Lateral skew of wheelset 1(blue) and wheelset 2 (green) under an applied input torque to wheelset 1.

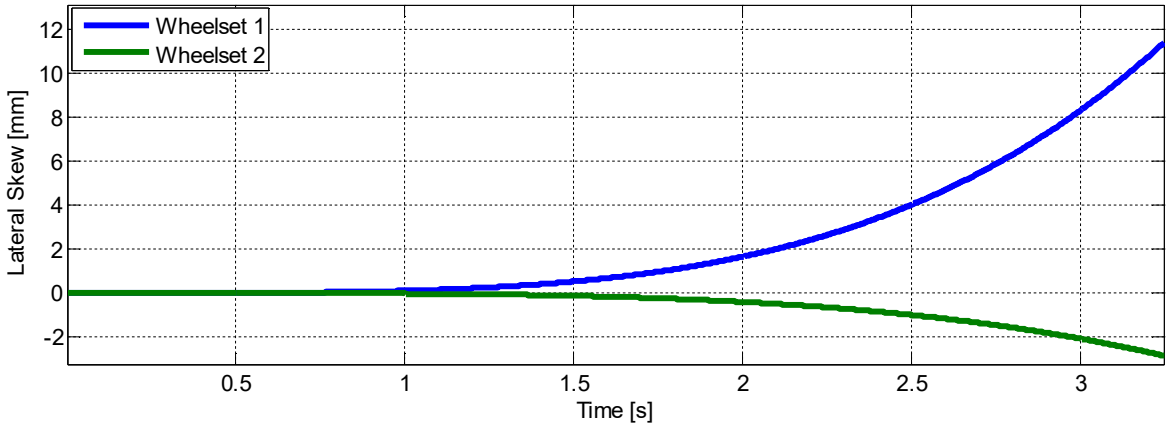


Figure 17– Simulation of the roller rig mathematical model (37)-(41) for IRW wheelset configuration. Lateral skew of wheelset 1(blue) and wheelset 2 (green) under an applied input torque to wheelset 1.

This behaviour would agree with our initial assumptions in the following way. As the wheel profiles are conical, should the rig start at a non-zero initial conditions a conventional wheelset would yield faster revolutions on one side of the wheelset as opposed to the other, due to the difference of rolling diameter of each respective wheel. This ultimately thus results in hunting motion resembling a harmonic output. Another interesting notion is the increasing amplitude of the produced harmonic output. This implies to us that the rig is already approaching its critical velocity to the brink of where the system is unstable at too high speeds of motion. As for IRW wheelsets, each wheel is allowed to run at a different rotational velocity, whereby lateral creeping is almost negligible, thus resulting in an almost constant behaviour from initial conditions. If an applied torque is given to such IRW wheelset, a clear veering can be seen leading away from the central axis of the bogie frame. Ultimately this veering is limited by a contact in the wheel flanges with respective rail heads and would thus introduce an element of nonlinearity.

5.2.1 HONU-MRAC Control Loop Design for Conventional Wheelsets

Following the proposed control loop design via HONU-MRAC, the proceeding section details its application to real-time active control as an IRW configuration on the CTU Roller Rig. To initiate investigation of the lateral skew control for the CTU roller rig, let us consider the state-space model from (37)-(41). In this example let us consider conventional wheelsets for a SISO system configuration, where only the process input $T_{\Psi 1}$ and corresponding process output y_{w1} are considered. Thus, we may consider application of state feedback design via the following modification of relation (36) as

$$\begin{aligned} \dot{x} &= (A - BK)x + Br \\ y &= Cx, \end{aligned} \tag{42}$$

where now

$$u = r - K \cdot x. \tag{43}$$

Here the variable r denotes the desired value and K is the state feedback gain vector. Then computation of the state feedback gain vector may be obtained as follows

$$T = R \cdot W, \tag{44}$$

where R is the controllability matrix. The shape matrix W is further given as

$$W = \begin{bmatrix} a_n & a_{n-1} & \dots & a_1 & 1 \\ a_{n-1} & \ddots & a_1 & 1 & 0 \\ \vdots & a_1 & 1 & \ddots & \vdots \\ a_1 & 1 & & & \\ 1 & 0 & \dots & & 0 \end{bmatrix}. \tag{45}$$

From (44), it yields that the controllability matrix R can be obtained as

$$R = [B \quad \dots \quad A^{n-1}B \quad A^n B]. \tag{46}$$

Following from the transformed state space relation

$$\begin{aligned} \overline{A} &= T^{-1} A T \\ \overline{B} &= T^{-1} B. \end{aligned} \tag{47}$$

The resulting state feedback gain vector yields to be

$$K = [\alpha_n - a_n, \alpha_{n-1} - a_{n-1}, \dots, \alpha_1 - a_1].T^{-1}, \tag{48}$$

where, a_i denotes coefficients of the characteristic polynomial for the matrix of dynamics A from (42). Further, α_i are coefficients corresponding to the n^{th} order polynomial equation

resulting from the desired pole placement. As a further, a HONU-MRAC is designed on the standalone mathematical model of the roller rig from relations (30)-(35). An LNU plant trained via RLS algorithm over 2000 epochs where the neural input vector features $n_y = 7$ previous model outputs and $n_u = 5$ previous process inputs, with extension of a QNU feedback controller of the same parameter configuration. Given relations (42)-(48), Figure 18 illustrates a comparison of the resulting SFC design

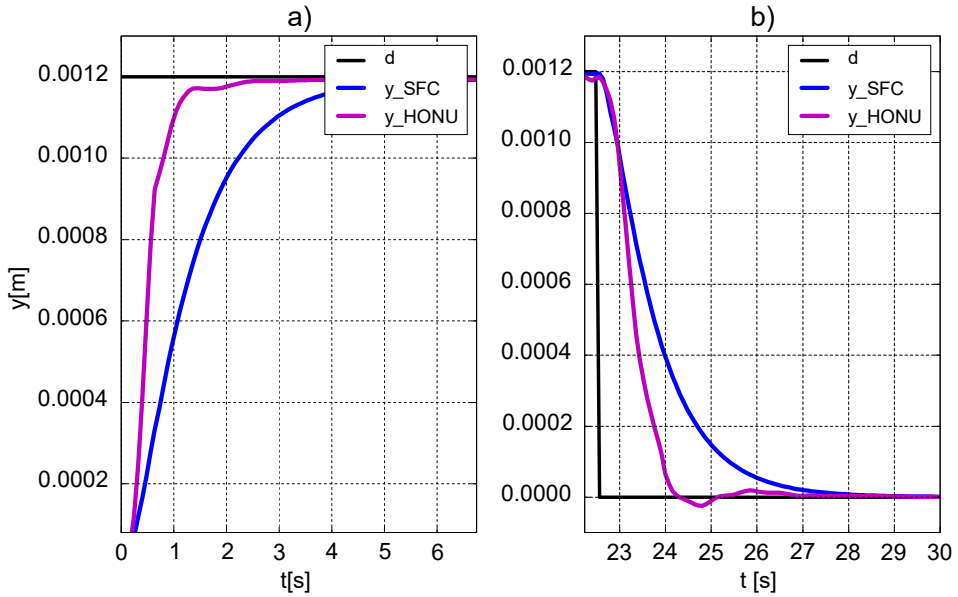


Figure 18– Step responses of the controlled bogie frame (30)-(35) with conventional wheelsets show superiority of LNU-QNU control loop (RLS $n_y = 7$, $n_u = 5$) over the achievable performance of SFC (in state feedback controller). Stability analysis via DHS and DDHS presented in Figure 52.

It is evident that the dynamic response of the HONU control loop is substantially faster than the SFC with similar capability to reduce the hunting motion of the leading wheelset. In the proceeding section, a further investigation into an independently rotating wheelset (IRW) configuration is presented.

5.2.2 IRW Active Wheelset Control via HONU-MRAC Configuration

Following the derived HONU-MRAC closed control loop as in Figure 18 a further topic is the investigation of an IRW wheelset configuration for active wheelset control. As can be seen in Figure 17 from a non-zero initial position, under an applied constant roller velocity or given a small constant input signal the IRW configuration veers the whole bogie frame in one direction which leads to the challenge of stabilizing the bogie system before enhancing control. From the presented HONU-MRAC closed loop in Figure 18, two control tasks are presented in this section. The first is to present the derived HONU-MRAC design for control

about a desired lateral skew setpoint. As a further, this dissertation introduces an effective approach to sliding mode control for stabilization of an IRW active wheelset, using the latter presented HONU decomposition. As the focus on this dissertation is primarily to investigate HONU-MRAC control design and HONU stability, this dissertation only introduces the sliding mode control concept via HONUs with remarks for future study.

Therefore, to initiate this study a numerical 3-D computer model via SIMPACK software with an extension pack to MATLAB Simulink is analysed via SISO configuration, where the control objective is to achieve a desired setpoint value of lateral skew on the leading IRW wheelset. The plant model in the MATLAB Simulink environment is initialized with a sampling period of 0.001[s] and considers the avoidance of wheel flange contact.

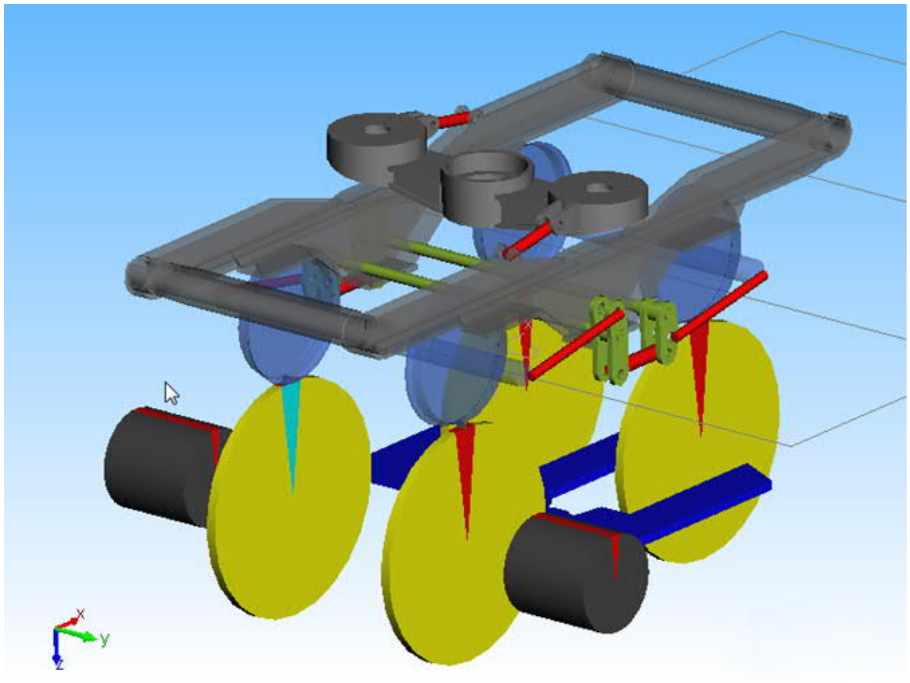


Figure 19 – 3D mechanical model of CTU Roller Rig used as plant for MATLAB Simulink HONU-MRAC closed control loop study, sampling rate=0.001[s] (Figure courtesy of the Dpt. of Automotive, Combustion Engine and Railway Engineering, FME CTU).

From the work [88], initial investigation via a linear state feedback controller (derived in analogy to the relations (42)-(48)) with additional cascade PID control showed that application on the linearized numerical model of the roller rig as in Figure 19 was only possible for continuous control loop stability with a sampling interval of < 1E-6. Thus, motivation further for an alternative approach of active control for IRW configuration. Given the above, a HONU- MRAC closed control loop is investigated.

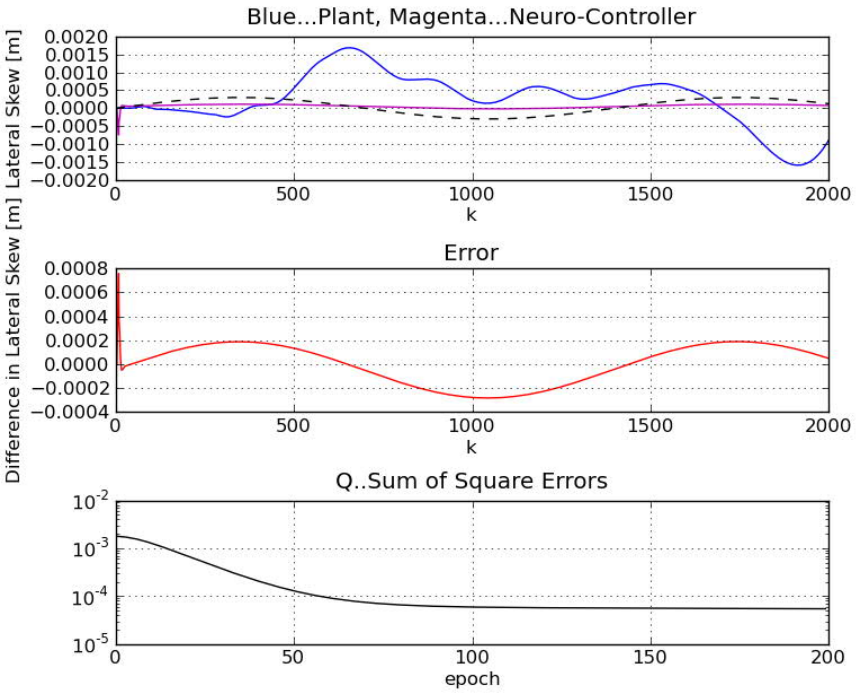


Figure 20 – Adaptive control of the 3-D mechanical model in MATLAB Simulink. LNU model trained via NGD with extension of LNU feedback controller trained via NGD where $n_y = 3$, $n_u = 3$ (as samples of error between the HONU output and setpoint) re-sampling = 5, epochs 200. As presented in [88], results generated from [98].

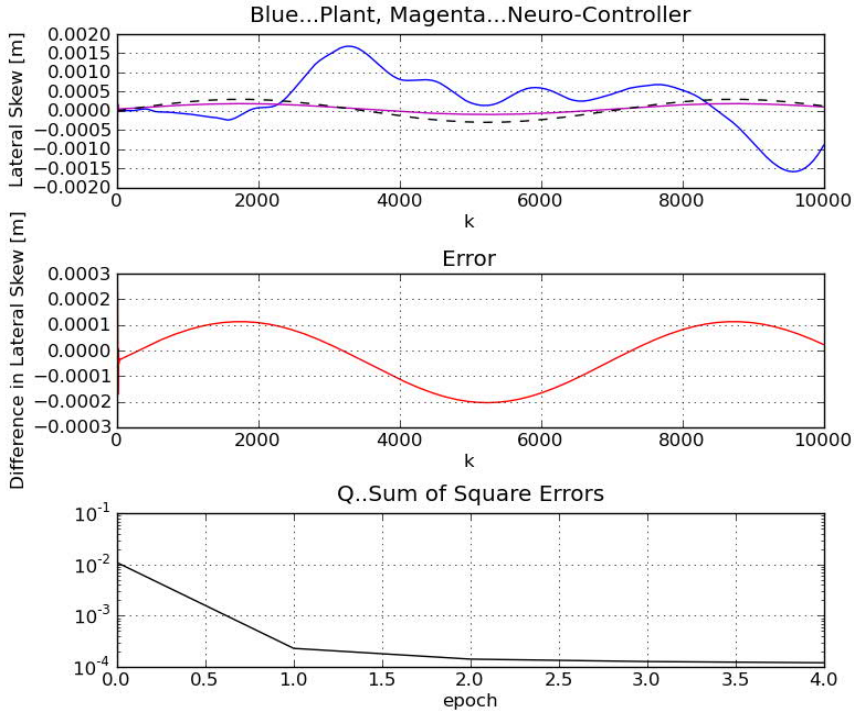


Figure 21 - Adaptive control of the 3-D mechanical model in MATLAB Simulink. QNU model trained via NGD with extension of QNU feedback controller trained via LM where $n_y = 3$, $n_u = 3$ (as samples of error between the HONU output and setpoint) re-sampling = 5, epochs 5. As presented in [88], results generated from [98].

A stable set of training data was derived with the standalone 3-D SIMPACK model (at mid-range speed) for various operating points in 0.001 sampling interval over 10 seconds. Given

adaptive identification via a QNU model featuring $n_y = 3$ step-delayed model outputs (lateral skew of the roller rig plant) and $n_u = 5$ step-delayed process inputs (yaw torque) with NGD learning, Figure 20 & Figure 21 show the performance of a derived HONU feedback control loop i.e. HONU MRAC closed control loop.

In this analysis Figure 20 illustrates a dynamic LNU model trained via the NGD supervised learning algorithm. A HONU feedback control of LNU architecture is then extended in feedback trained over 200 epochs of the NGD algorithm. The configuration chosen features $n_y = 3$ step-delayed model outputs (representing the lateral skew of the roller rig plant) and $n_u = 3$ step-delayed process inputs as samples of error between the HONU output and setpoint value. It is evident in the upper plot of Figure 20 where the black-dashed line depicts the setpoint value, that the HONU-MRAC closed control loop as a standalone features a desirable control performance in adhesion to the desired behaviour of the lateral skew. As depicted in Figure 21 only 5 epochs of the supervised learning algorithm are necessary to reduce the SSE to an order of 10^{-4}. Using the same configuration parameters in the QNU feedback controller of $n_y = 3$ step-delayed model outputs and $n_u = 3$ step-delayed process inputs as samples of error between the HONU output and setpoint value, the plot of errors highlight the superior performance of the QNU-QNU closed control loop configuration. Therefore, from this experimental study on a 3-D mechanical model, the proceeding section presents implementation of such HONU-MRAC control loop design on the real CTU Roller Rig system. Due to the nature of the control problem, a further appropriate study is the application of a sliding mode based HONU control, as deep study of such approach is out of the scope of this dissertation only preliminary results and discussion are presented in section 10.2 for IRW active wheelset control.

5.2.3 HONU-MRAC Design on Real Roller Rig with IRW Configuration

Following the observation in Figure 17 for an IRW wheelset configuration given no further control action, once the leading wheelset beings to steer towards a given side of the running track the rig veers to the point of wheel flange contact which in the real system is $\pm 6[mm]$ from the central axis. Therefore instead of an adequately quick change of the yaw torque as applied in Figure 20 & Figure 21, as an initial investigation, a conventional P-controller is introduced in extension to the existing system as a means of initial control. Figure 22 (blue

line) illustrates the response of the roller rig under a tuned P-control loop. In this study, the roller rig is set to have a constant speed of 60 rpm. From this result we may observe the clear steady state error deviation that the control loop exhibits from the standalone P-control loop.

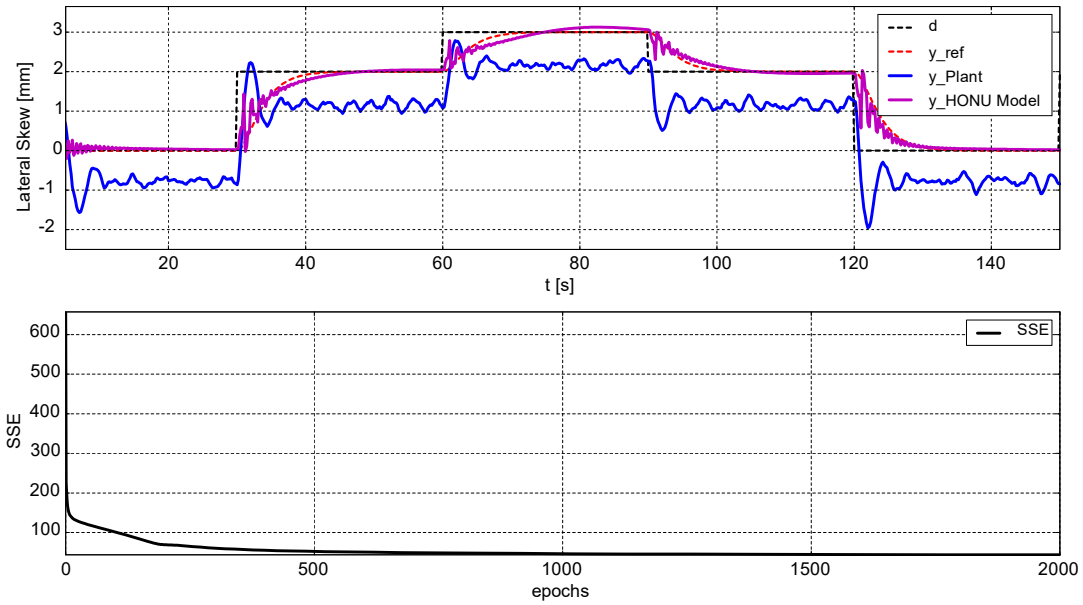


Figure 22 – IRW wheelset of the Roller Rig at 60 rpm with HONU-MRAC adaptive control loop via LNU model with QNU feedback controller based on GD learning algorithm. $n_y = 4$, $n_u = 5$ and adapted feedback gain =1.11 as constant parameter control loop. As presented in [95] and concepts in [88].

Further, to this a clear oscillatory wise behaviour may be exhibited when the roller rig reaches a steady setpoint values, from the remarks of Figure 16 & Figure 17 this is described as hunting motion, where due to the opposing side wheels for one wheelset being on different conical diameters the roller rig has a rotational tendency to veer about the mean axis of the desired setpoint value. Therefore, from Figure 22 it is apparent that although a large degree of steady-state error is evident in with use of the standalone P-closed control loop, as an initial foundation the control problem may be shaped to investigate controller optimisation. Thus, the design objective may be to further minimise steady state error and reduce hunting motion of the roller rig under active steering control. Figure 22 therefore illustrates the first proposal of a HONU-MRAC design. Here, a first-order HONU (LNU) is used to adaptively identify the dynamic response of the roller rig under proportional feedback control. In lieu a QNU feedback controller is added in extension with a respective plant reference model. In this configuration both the HONU model and respective feedback controller feature 4 previous samples of the process output ($n_y = 4$) and 5 previous samples of the process input, in this case being the desired value of the roller rig ($n_u = 5$), trained via GD algorithm over 200 epochs. Figure 23 depicts a second-order HONU (QNU) model with extended QNU

feedback controller. In this configuration, 4 previous samples of the process output ($n_y = 4$) and 5 previous samples of the desired value of the roller rig ($n_u = 5$) are used. Furthermore, a batch form of training (LM algorithm) under the same model parameter settings reveals greater reduction in square error in comparison to the previous LNU architecture which corresponds with smoother transition in the HONU-MRAC response about each step-wise change of desired value.

Figure 24 & Figure 25 show the resulting online implementation of a HONU–MRAC control loop based on both an LNU and QNU model with QNU controller for both the GD and LM based algorithms respectively. Here both executions are applied as a feedback controller of constant parameters directly implemented from the tuned results in Figure 22 & Figure 23. It is intriguing from these results that although the LM algorithm delivered better performance following offline tuning, during online application it is in fact the GD trained LNU model with QNU controller that yields the overall best performance amongst all investigated configurations. Comparison of the mean absolute error (MAE) in Table 8 shows minimal deviation in steady state error via GD learning in comparison to the LM based controller. A reason for this may be due to the GD algorithm being better in mapping the contemporary dynamics of the engineering process, as during its respective model training by nature of the GD algorithm incorporates more of the roller rig hunting motion behaviour.

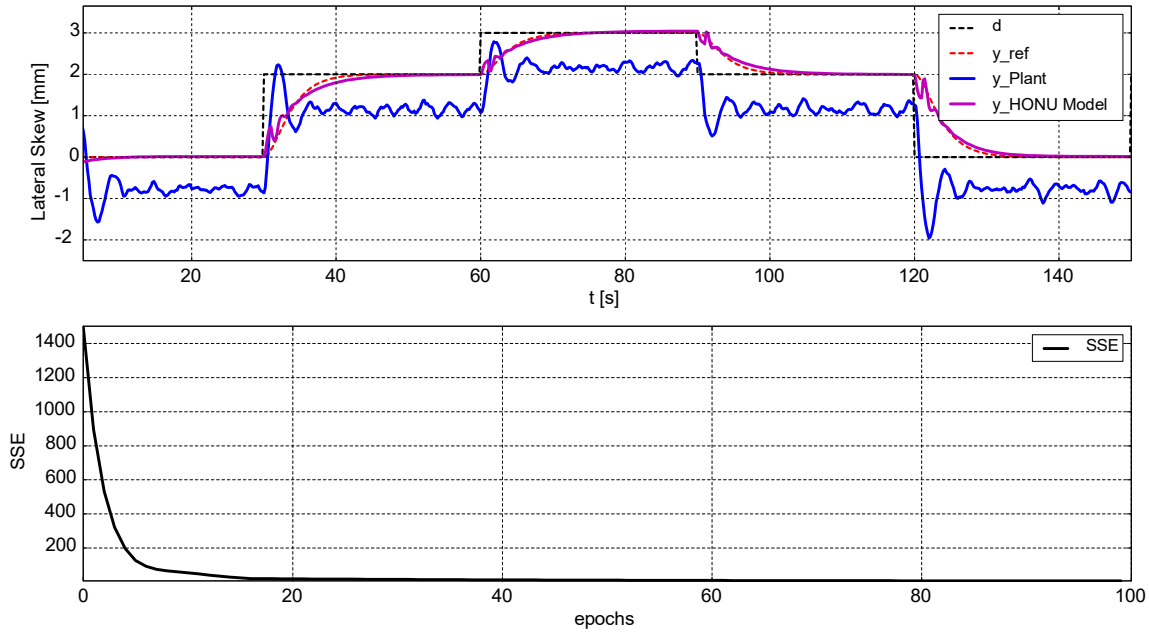


Figure 23 - Theoretically tuned HONU-MRAC adaptive control loop via QNU model with QNU feedback controller based on LM learning algorithm. $n_y = 4$, $n_u = 5$ and adapted feedback gain =1.022 as constant parameter control loop. As presented in [95] and concepts in [88].

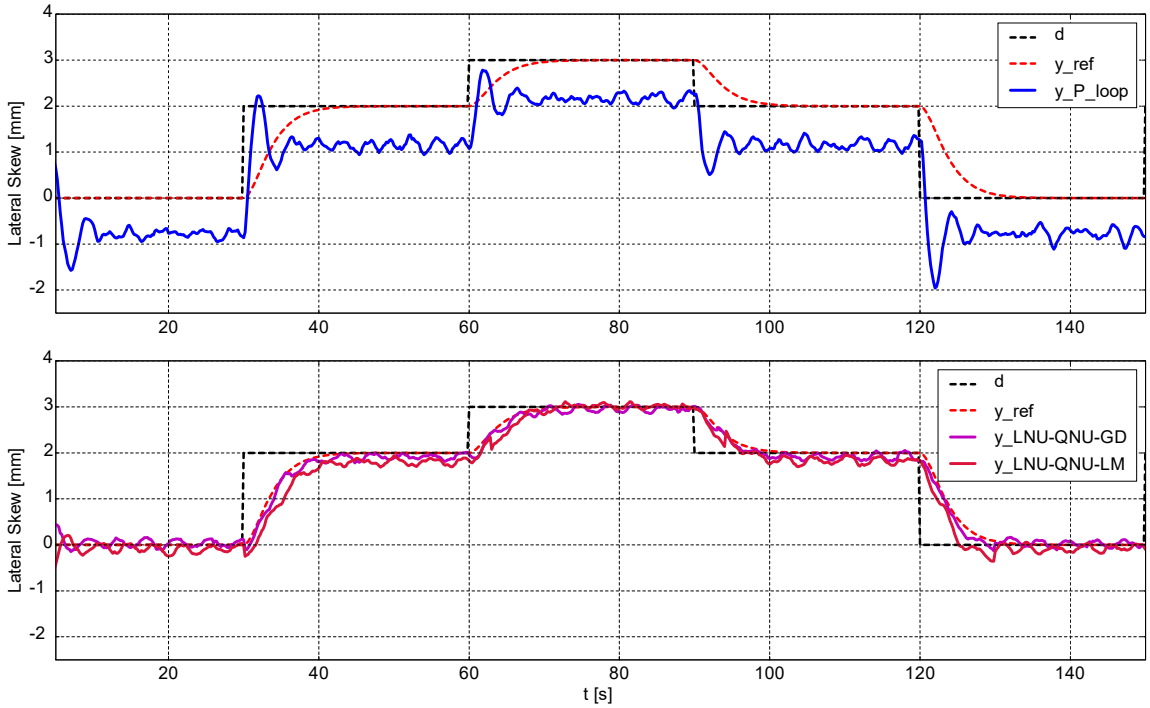


Figure 24 – Real-time HONU-MRAC adaptive control loop via LNU model with QNU controller based on GD and LM learning algorithm. $n_y = 4$, $n_u = 5$, *as constant parameter control loop. As presented in [95] and concepts in [88]. Stability analysis via DHS presented in Figure 53 & via QNU-LNU configuration, DDHS analysis in Figure 54.*

.

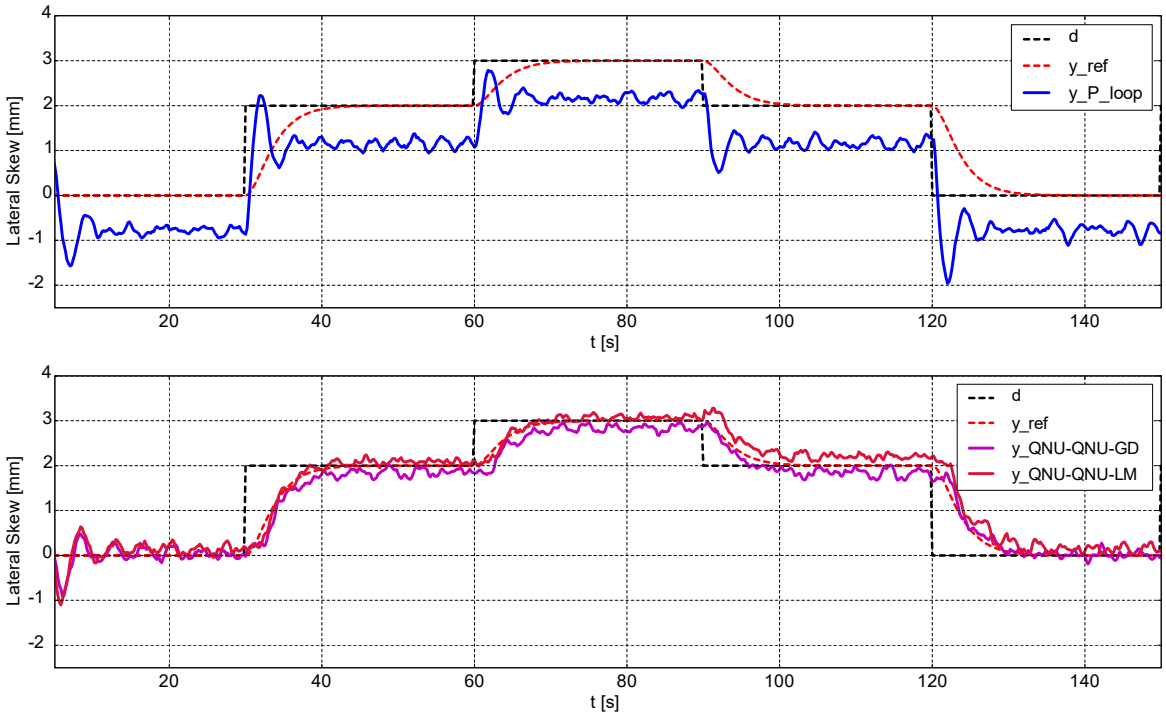


Figure 25 - Real-time HONU-MRAC adaptive control loop via QNU model with QNU feedback controller based on GD and LM learning algorithm. $n_y = 4$, $n_u = 5$ *as constant parameter control loop. As presented in [95] and concepts in [88].*

TABLE 4 PARAMETERS OF HONU-MRAC CONSTANT PARAMETER CONTROL LOOP CONFIGURATIONS				
HONU Plant Model	**HONU Controller Training**	**No. HONU Model Outputs**	**No. of Desired Value Inputs**	**MAE**
LNU_QNU	GD	4	5	0.090656501581
LNU_QNU	LM	4	5	0.172675627428
QNU_QNU	GD	4	5	0.215209570572
QNU_QNU	LM	4	5	0.242736610099

Following the preceding analysis of HONU-MRAC performance for different configurations as a constant parameters control loop, during the latter part of 2016 changes to the bogie frame design on the Roller Rig, influencing the bogie frame stiffness and damping properties have caused significant deterioration in the previously designed control loop performance. Thus, a further goal from this section is to analyse the ability of the HONU-MRAC adaptive control to adjust the derived controller parameters for optimisation under such mechanical property changes in the engineering system. Figure 26 illustrates the performance of a fully adaptive HONU-MRAC closed control loop.

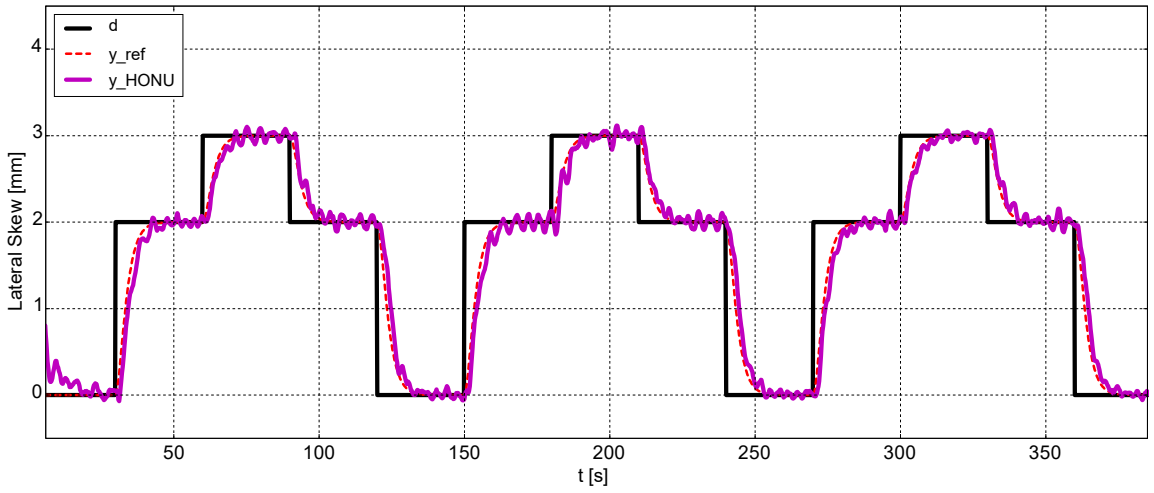


Figure 26- Fully adaptive HONU-MRAC control loop via an adaptive LNU model trained via RLS with adaptive QNU feedback controller trained via the NGD learning algorithm, with GD pre-training as identified in Figure 22 ($n_y = 4$, $n_u = 5$). Concepts presented in [95]and [88].

In this setup, both the LNU model and QNU feedback controller are adaptively tuned in real-time. The original GD trained HONU-MRAC control parameters are used as initial conditions for the online adaptive update rules. The LNU HONU model is trained via the RLS learning algorithm where the adaptive QNU feedback controller is tuned via the NGD algorithm. Following this approach, Figure 26 depicts in the first 150[s] of application that the HONU-MRAC control loop indeed achieves desirable control performance immediately

adhering in just the initial samples of application to the required process setpoint. Additionally, the real time adaptive algorithm is able to further minimise the deviations in error caused by hunting motion after 3 epochs of process data (seen from 250[s] in Figure 26). In Figure 54 a) the previously derived QNU model is applied on the new bogie frame with changes stiffness and damping properties. In this example an adaptive NGD training algorithm is used, where the LNU feedback controller is trained via an adaptive RLS algorithm. As illustrated, after 4 epochs of the process data the HONU-MRAC closed control loop is able to minimise the steady state error substantially which exhibits the capability of such QNU-LNU control loop for fast real-time adaptive tuning for control loop optimisation.

5.3 *HONU-MRAC Design for Barrier Drive Control*

Another application in the field of rail automation concerns motor drive control for a redesigned barrier drive system. The ultimate aim of this presented case study is to analyse the existing drive control and propose an enhanced design which could be applied as firmware to the drive control board for control of a level crossing barrier arm system. Due to sensitivity of unpublished terms and details of the presented application, the terminology used in this chapter of this dissertation work has been generalized in accordance to the presented work [96].

The existing drive control board communicates with a main industrial controller via standardized Ethernet protocols to communicate and read status data from the drive unit. A key requirement for operation in SIL4 safety level crossing systems is that transition time (of either 6 or 10 seconds) and prescribed voltage and current limiting values of the design are maintained. From initial tests factors such as the span of the barrier boom arm, furthermore dynamical changes with respect to the wear of the respective motor bearings, tensioning of the barrier belt. In addition, environmental factors like build-up of snow during the winter months, or unforeseen loading of the barrier boom due to human influence can cause an exceed of this required time limit as well as an undesirable motor speed characteristic. As a result, the necessity for design of a control loop, capable of maintaining the raising and lowering time tolerances whilst, ensuring the motor current and voltage limits in accordance to the system requirements are maintained, is presented. Due to validation of the existing prototype, an extension to the existing control loop design is the most sought-after approach. Due to such influencing factors presented, adaptive control yields to be a rather advantageous method for solution of this barrier drive control problem.

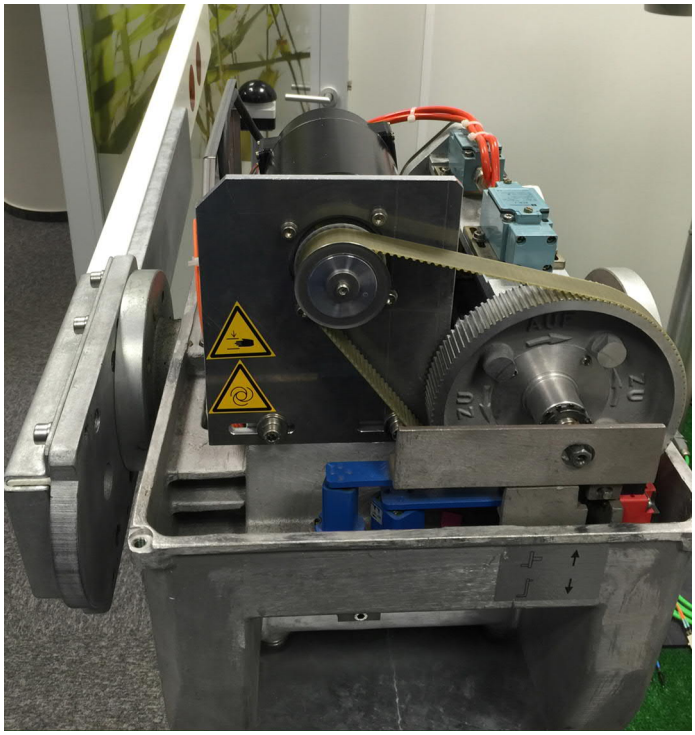


Figure 27 – Barrier arm system drive with prototype supporting structure and barrier drive control board. As presented in [96]. See the control results in Figure 32 & Figure 33 and stability analysis via DHS in Figure 55 and DDHS in Figure 57.

Thus, in this dissertation work, an investigation into the application of a HONU-MRAC control loop will be studied. Recent applications of the control board firmware, feature an existing cascade PID control loop, applied with constant parameterisation of the respective controller parameters, due to this such mechanical changes or deviations in the barrier boom load yield a limited degree of application for the derived controller parameters.

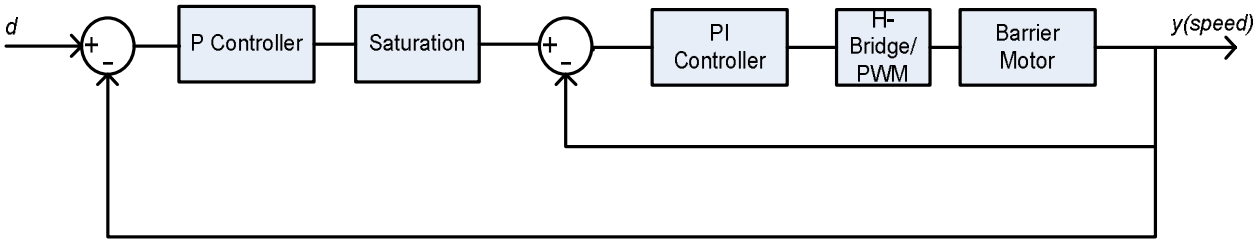


Figure 28 – Existing cascade P with PI closed control loop implemented on prototype BRU for SIM 6/13 barrier motor control.

Figure 28, illustrates the existing implementation a cascade P control loop with PI controller as applied to the barrier drive. The motor itself is a GROSCHOPP PM6 114-70 DC brushed motor, which may be controlled via a H-bridge and pulse width modulation (PWM) in order to regulate the voltage supplied the motors armature. An included saturation is incorporated in order to limit the current of the motor within range of ±8000[mA]. Figure 29 shows the respective output measurements taken from one open/close cycle of the barrier arm. From

these measurements a rather intriguing characteristic may be found in the measured current curves. In the initial stages of application, the barrier drive holding magnets are released where an internal holding spring at the end positions of the barrier arm momentarily propagates the arm forward. This is translated in the motor as an opposing sign of current flow hence acting as a generator. Following these initial samples, the current across the motor armature rises to a constant until the braking phase of the barrier drive is initiated. In this braking phase, the barrier arm reaches the opposite end position and now the end position spring acts to retard the arm, further pronounced by the physical property of reduced motion at each end of the linear motion screw coupled to the barrier motor which is reflective in the current characteristic as a further increase. This characteristic is thus too translated into the respective armature voltage curves. In addition, from the measured data a large spike in the motor current is evident in vicinity of transitions at activation and deactivation of the barrier. A reason for this is largely due to the instantaneous switching of the motor H-bridge and is thus independent of the dynamics of the barrier motor itself.

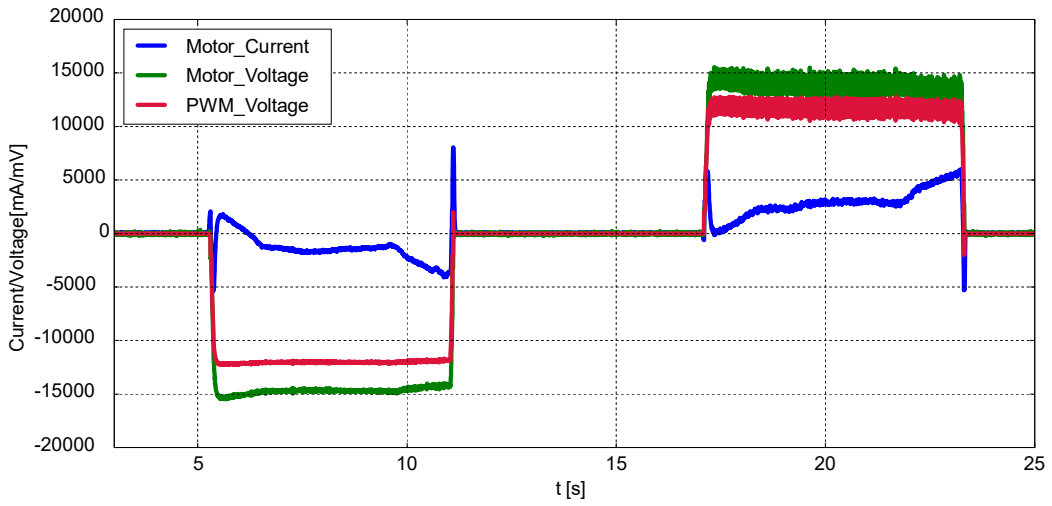


Figure 29 – Current and voltage measurements of barrier drive 1 for one opening and closing cycle, used for deriving the process output value of barrier motor via the relation (49).

From the measurements depicted in Figure 29, it is the speed of rotation of the barrier motor drive in which we are interested to control which may be derived via the following relation

$$k_c \Phi_{MOT} v = \frac{1}{\varphi}[V_{MOT} - I_a R_{MOT}], \tag{49}$$

where here k_c is the motor load constant, Φ_{MOT} is the magnetic flux constant [Wb], v is the overall speed of the motor [rpm], V_{MOT} [mV] is the voltage across the barrier motor armature, I_a [mA] is the measured armature current and R_{MOT} [ohms] denotes the internal

motor resistance and the variable φ serves as a conversion constant with respect to the unit of measure. From this relation, Figure 30 depicts the corresponding speed of the barrier motor during one open/close cycle for two respective barrier drive systems. The barrier drives used are identical in construction as well as the tuned cascade PI controller parameters via their loaded firmware; however, barrier drive 2 is mounted with a loaded 4[m] arm on the motor output shaft. Another interesting characteristic important to note is the behaviour of speed at the barrier end positions. From both the speed profile of the barrier arm during a raising and lowering phase, in the initial moments of application an overspeed is shown. As is reflective in the motor current curves, this is due to the internal spring at the end positions of the barrier arm release pushing the arm forward with a rapid force due to release of the barrier holding magnets. Following these initial samples, as voltage is supplied to the motor armature the speed of the motor plateaus to a constant speed before the deceleration phase is initiated encroaching the desired end position of the barrier arm. Here too the opposite side end position spring acts to retard the motion of the barrier arm and hence causes a reduction in speed before the motor is braked via a sudden deceleration to a powered off state.

As can be seen from t=18 [s] onwards in Figure 30 b) the corresponding speed of the barrier drive 2 is lower than that of barrier drive 1 in Figure 30 a) resulting in a longer time of travel for the barrier to fully lower or raise within the prescribed time of transition t_{trans}. This behaviour is predominantly a problem with introducing larger span barrier arms with substantially higher loading on the barrier motor. Further factors like deviating mechanical properties due to wear may also cause such significant deviations. With regards to the barrier drive 2 response, a problem is presented as the design of the barrier control must withhold a prescribed time of motion so called “motor protection time” in order claim safe securing of the level crossing. As a result, this defines the underlying problem of the current control setup, where the cascade PI controller parameters are not universally applicable for any type of barrier drive and must be re-tuned due to mechanical property deviations or use of different boom arm lengths. Following analysis of the data in Figure 30, the action of the end position springs in tie with the linear motion screw behaviour of the barrier arm on its end positions, results in a nonlinear characteristic which motivates the use of a nonlinear HONU model. Given this, a QNU is chosen and derived for control optimisation of the given barrier drive process data in Figure 30 in a bid to provide a more universally applicable controller for the applied barrier drive motor.

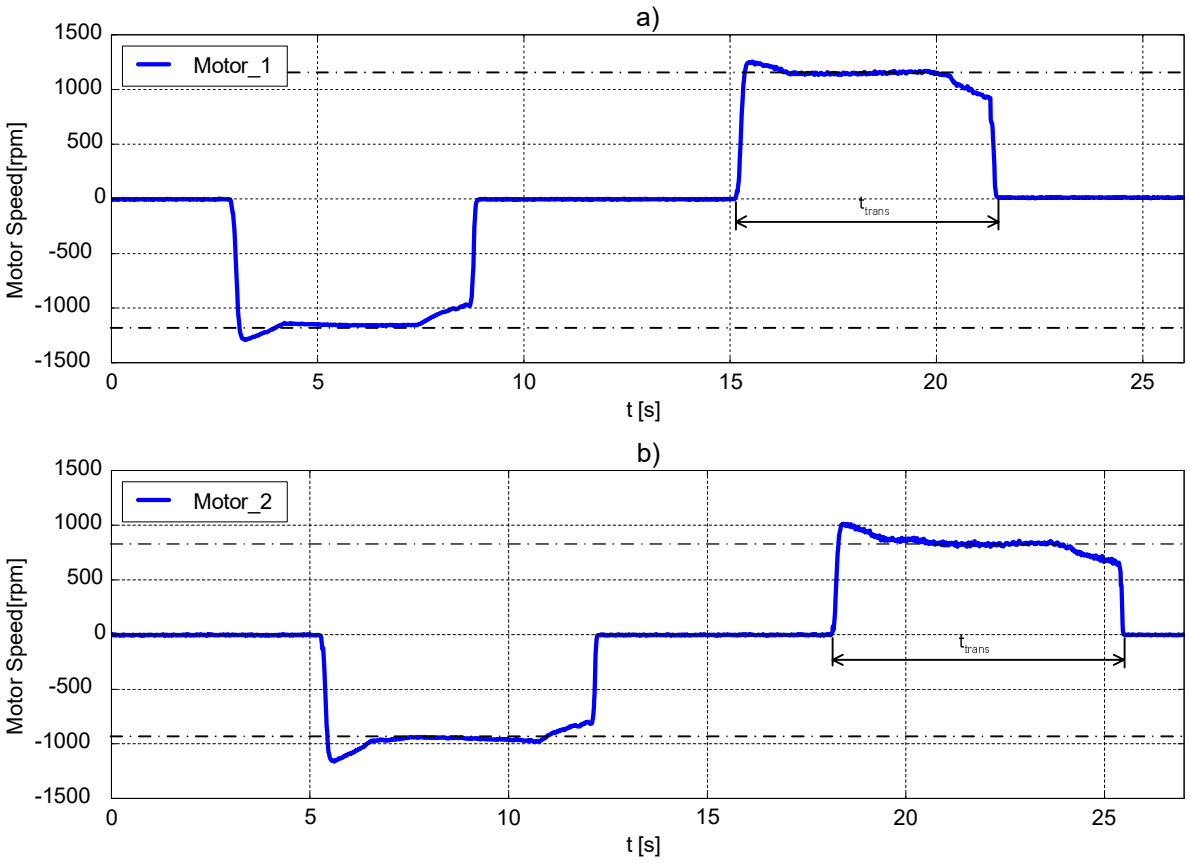


Figure 30– Cascade P with PI closed control loop response of barrier motor for one closing and opening cycle derived from (49) for two different barrier drives. a) Barrier Drive 1: t_{trans} 6[s] b) Barrier Drive 2 (with loaded 4m boom arm): t_{trans} 7.1[s] with same controller parameter settings.

As an initial a QNU featuring 3 previous outputs of the HONU model n_y and 5 previous inputs n_u is used, trained via the RLS algorithm over 50 epochs for plant identification of the barrier motor during one cycle for both respective drives. Following this, the GD algorithm is used for training a QNU feedback controller over 80 epochs, incorporating the same length of step-delayed outputs as the identified HONU model. With regards to the desired process dynamics during deceleration of the barrier motion, as the end position springs indeed cannot be freely controlled nor omitted from the barrier arm construction the chosen reference model is configured such to incorporate this deceleration component in contrast to the constant setpoint used in the initial cascade PI control loop design.

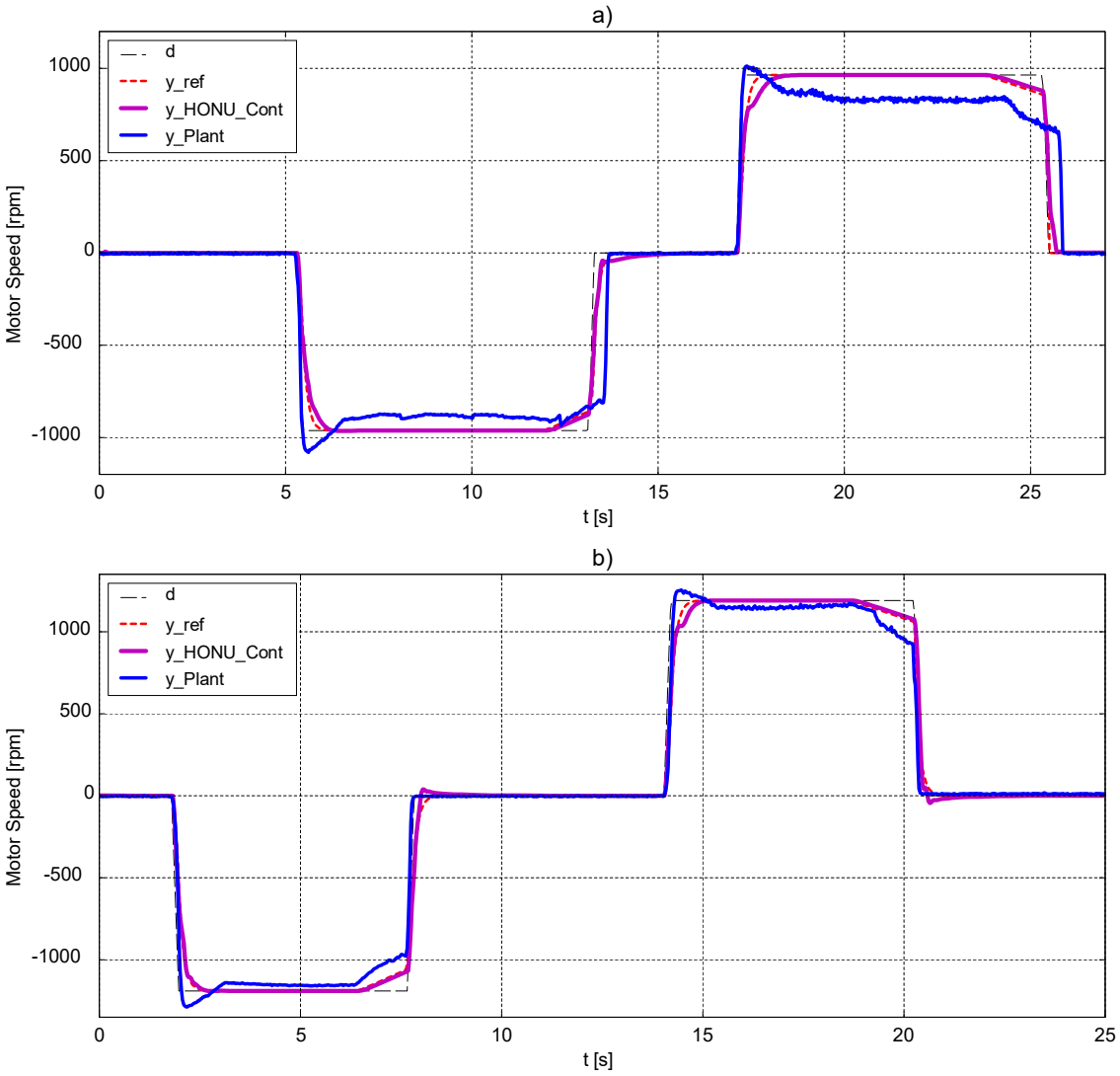


Figure 31 – Offline tuned HONU-MRAC control loop, trained for both barrier drives a) Barrier drive 2 b) Barrier drive 1. Here a QNU trained via RLS algorithm ($n_y = 3$, $n_u = 5$) for plant identification is used with QNU trained via GD as a HONU feedback controller ($n_y = 3$, $n_u = 5$).

In Figure 30 a), the barrier motion in one direction is configured for an overall duration of 6 seconds with the existing cascade closed control loop operating at a sampling interval of 0.004 [s]. For the derived HONU-MRAC control loop the sampling interval of 0.04 [s] is used serving as an adequate compromise between optimal adaptive identification of the HONU-MRAC control loop and application on the real drive control board. Figure 31 thus illustrates the performance of the derived HONU-MRAC control loop following the respective parameters and applied training algorithms. In Figure 31, after 80 epochs the sum of square errors (SSE) reaches a minimum with the simulated controller output adhering to the desired setpoint across both operational states of the barrier motion quite successfully. Furthermore, the offline tuned HONU-MRAC control loop response yields similar performance for the rather less optimal cascade PI control loop data corresponding to barrier

drive 2 Figure 31 a) as compare with the more adequately performing response on application for barrier drive 1 Figure 31 b). From these results, Figure 32 therefore illustrates the application of the derived HONU-MRAC control loop on the real barrier drive system with Figure 33 illustrating the real-time control loop application derived in Figure 31 b).

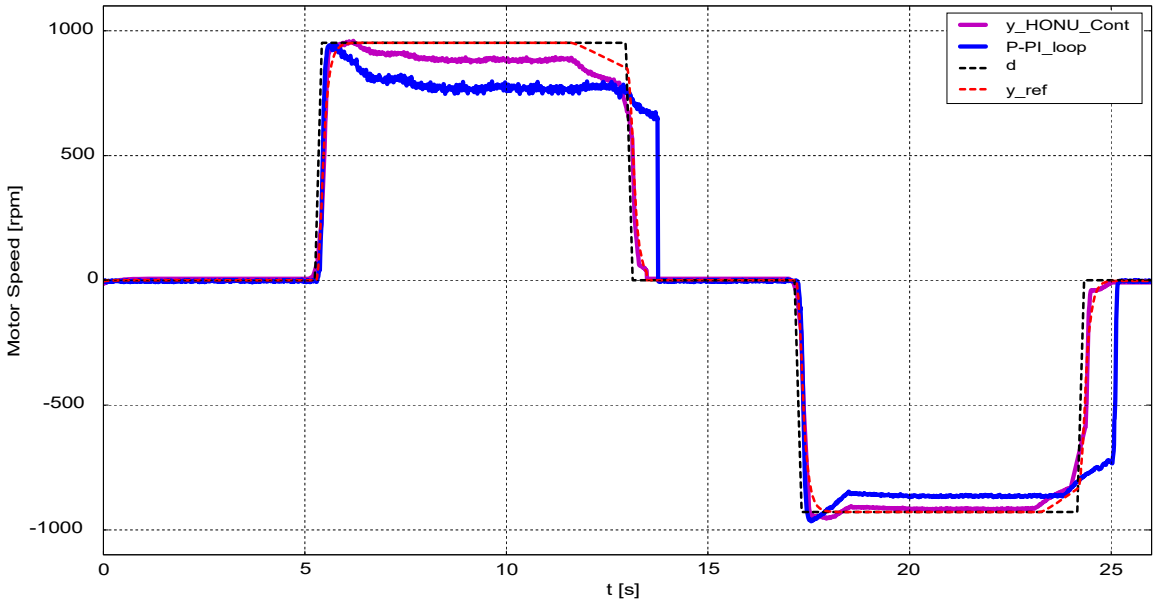


Figure 32 – Barrier drive 2 (loaded 4m barrier arm): Real-time HONU-MRAC control of barrier drive motor for one opening and closing cycle. QNU-QNU ($n_y = 3$, $n_u = 5$) identified via RLS as a plant followed by GD learning for the HONU feedback controller. Superior performance is achieved in comparison to standalone cascade control loop Figure 30. Stability analysis via DHS and DDHS respectively presented in Figure 55 & Figure 57.

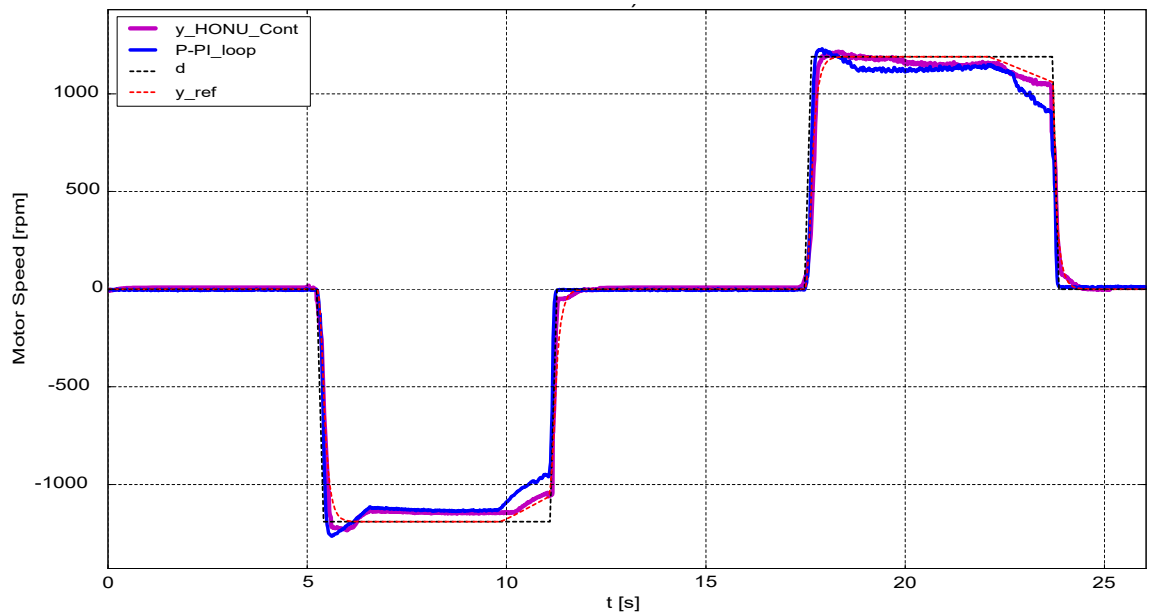


Figure 33– Barrier drive 1: Real-time HONU-MRAC control of barrier drive motor for one opening and closing cycle. QNU-QNU ($n_y = 3$, $n_u = 5$) identified via RLS as a plant followed by GD learning for the HONU feedback controller. HONU-MRAC control loop performs more optimally in comparison to standalone cascade control loop for both respective drives.

As may be compared the resulting output of the closed cascade control loop with extension of the HONU-MRAC adaptively tuned control loop yields very similar results in terms of its control performance with respect to the desired setpoint of the barrier motor speed during both in an opening and closing phase. On comparison of this result with Figure 30, the resulting steady state error under application of the HONU-MRAC is substantially reduced with respect to the standalone closed cascade control loop, whilst maintaining the required motor protection time span for motion of the barrier boom. What yields to be further intriguing is the performance of the derived HONU-MRAC closed control loop on real-time application to barrier drive 1. From Figure 33, a slightly more optimal response is achieved on application of the HONU-MRAC control loop, primarily in vicinity of the initial overshooting and deceleration phase. Therefore, it may be concluded that the derived HONU-MRAC control loop applied as a constant parameter feedback control loop design in real-time yields more optimal performance than the standalone cascade PI control loop. What may be concluded from this application, is the strength of nonlinear HONU architectures for adapting to variable behaviour of the barrier drive action under different loading conditions to provide a universal set of controller parameterisation. Although, this dissertation studied two loading cases this principle may be extended for more loading cases of expected barrier arm lengths and tested for correct identification and control as a constant parameter controller, or as an adaptive solution for re-tuning during an official maintenance procedure.

Throughout this section, the advantages of HONU-MRAC control loop design were highlighted as seen, with low to moderately nonlinearities such as variable static plant gain, or linear oscillatory systems. The HONU-MRAC was able to not only provide desirable steady state performance, but also optimisation in dynamic response of convergence for the presented system examples. As with any control loop design, another challenge is to justify the derived approach is stable especially with an already existing control law not derived from stability theories e.g. Lyapunov function criteria. Therefore, the main contribution of this dissertation is presented in the proceeding sections to derive two novel approaches for pointwise state-space based representation of HONU architectures and further their BIBO and BIBS stability evaluation as a standalone and as a MRAC based control loop.

6 DHS: Discrete-Time Higher Order Neural Unit Stability

Following overview of HONU architectures and their fundamental supervised learning algorithms along with several presented examples of their real implementation, this section

covers a natural first step in finding a core relation between HONU architectures and classical non-linear discrete time state space models. This section derives a novel linearized pointwise state-space representation for HONU models and their closed MRAC based control loop which ultimately leads to a BIBO based stability condition termed as DHS (Discrete-Time HONU stability). Due to the approached being centred around a locally derived, pointwise linearized state-space model the approach is presented for lower order polynomial architectures and where the proceeding section then looks at a novel decomposition approach for representation of higher order polynomial models that also accounts for the time-variance of process dynamics. Throughout experimental analysis, DHS method is also used to justify functionality of the proposed DDHS method and highlight their respective advantageous on the applied engineering applications.

6.1 Pointwise State-Space Representation of Dynamic Nonlinear Neural Unit Models

As exhibited in the preceding section, for certain engineering applications it is advantageous to utilise a nonlinear HONU model due to its superiority in mapping nonlinear input output relations as well as faster learning response with supervised learning algorithms as such that of GD, RLS or L-M training algorithm. As was presented in [90] the underlying polynomial structure of a HONU draws an intrinsic analogy to discrete-time dynamic state space models, therefore this section investigates the transformation of a quadratic HONU (QNU) model into state-space representation from known nonlinear system state-space model theory [44], [84].

To develop this approach, consider the general expansion of a QNU where an arbitrary length of previous model outputs n_y, further previous process inputs n_u are considered and for simplification $n_{\bar{x}} = n_y + n_u$,

$$\begin{aligned}\tilde{y}(k) = \sum_{i=0}^{n_x}\sum_{j=i}^{n_x}\mathbf{W}_{i,j}x_i x_j = w_{0,0} + w_{0,1}\tilde{y}(k-1) + w_{0,2}\tilde{y}(k-2) + \ldots + w_{0,n_y}\tilde{y}(k-n_y) \\ + w_{0,n_y+1}u(k-1) + \ldots + w_{0,n_{\bar{x}}}u(k-n_u) + w_{1,1}\tilde{y}(k-1)\tilde{y}(k-1) + \ldots + w_{1,n_y}\tilde{y}(k-1)\tilde{y}(k-n_y) + \\ w_{2,2}\tilde{y}(k-2)\tilde{y}(k-2) + \ldots + w_{n_{\bar{x}},n_{\bar{x}}}u(k-n_u)u(k-n_u).\end{aligned} \tag{50}$$

On observation of (50) if we wish to construct such quadratic, nonlinear polynomial model into state-space representation in terms of its parameters we may select the following formulation of state variables corresponding to the state variable vector $\bar{\mathbf{x}}$

$$\begin{aligned}\bar{x}_1(k) &= \tilde{y}(k-n_y+1)\\ \bar{x}_2(k) &= \tilde{y}(k-n_y+2)\\ &\vdots\\ \bar{x}_{n_y}(k) &= \tilde{y}(k).\end{aligned} \tag{51}$$

Given (51) we may express the above state variables over the next sample as follows

$$\begin{aligned}\bar{x}_1(k+1) &= \bar{x}_2(k)\\ \bar{x}_2(k+1) &= \bar{x}_3(k)\end{aligned} \tag{52}$$

$$\begin{aligned}&\bar{x}_{n_y}(k+1) = w_{0,0} + w_{0,1}\bar{x}_{n_y}(k) + w_{0,2}\bar{x}_{n_y-1}(k) + w_{0,n_y+1}u(k) + .. + w_{0,n_{\bar{x}}}u(k-n_u+1)\\ &+ w_{1,1}\bar{x}_{n_y}(k)\bar{x}_{n_y}(k) + w_{1,2}\bar{x}_{n_y}(k)\bar{x}_{n_y-1}(k) + ... + w_{n_{\bar{x}},n_{\bar{x}}}u(k-n_u+1)u(k-n_u+1).\end{aligned} \tag{53}$$

From the relation (53) due to the introduced nonlinear input term, an affine-control formulation of the input states can be applied via introduction of the state-variable term $u(k)$ and replacement with the additional integrator input term $\bar{u}(k)$ as follows

$$\bar{x}_{n_y+1}(k) = u(k-n_u+1), \tag{54}$$

from (54), we may restructure the state equations of (52)-(53) to then be

$$\bar{x}_1(k+1) = \bar{x}_2(k) \tag{55}$$

$$\begin{aligned}\bar{x}_2(k+1) &= \bar{x}_3(k)\\ &\vdots\end{aligned} \tag{56}$$

$$\begin{aligned}&\bar{x}_{n_y}(k+1) = w_{0,0} + w_{0,1}\bar{x}_{n_y}(k) + w_{0,2}\bar{x}_{n_y-1}(k) + w_{0,n_y+1}\bar{x}_{n_y+1} + w_{1,1}\bar{x}_{n_y}(k)\bar{x}_{n_y}(k)\\ &+ w_{1,2}\bar{x}_{n_y}(k)\bar{x}_{n_y-1}(k) + ... + w_{n_{\bar{x}},n_{\bar{x}}}\bar{x}_{n_{\bar{x}}-1}\bar{x}_{n_{\bar{x}}-1}.\end{aligned} \tag{57}$$

$$\bar{x}_{n_{\bar{x}}}(k+1) = \bar{u}(k). \tag{58}$$

From relations (55)-(58), the respective state equations may then be expressed into state-space representation in the following manner,

$$\begin{bmatrix}\bar{x}_1(k+1)\\ \bar{x}_2(k+1)\\ \vdots\\ \bar{x}_{n_y}(k+1)\\ \vdots\\ \bar{x}_{n_{\bar{x}}}(k+1)\end{bmatrix} = \begin{bmatrix}\bar{x}_2(k)\\ \bar{x}_3(k)\\ \vdots\\ w_{0,0} + w_{0,1}\bar{x}_{n_y}(k) + w_{0,2}\bar{x}_{n_y-1}(k) + w_{0,n_y+1}\bar{x}_{n_y+1} + w_{1,1}\bar{x}_{n_y}(k)\bar{x}_{n_y}(k)\\ + w_{1,2}\bar{x}_{n_y}(k)\bar{x}_{n_y-1}(k) + ... + w_{n_{\bar{x}},n_{\bar{x}}}\bar{x}_{n_{\bar{x}}-1}\bar{x}_{n_{\bar{x}}-1}\\ \vdots\\ 0\end{bmatrix} + \begin{bmatrix}0\\0\\0\\0\\ \vdots\\ \bar{u}(k)\end{bmatrix}, \tag{59}$$

$$\bar{y}(k) = \bar{x}_{n_y}(k). \tag{60}$$

Thus, from relations (59)-(60), a general form for the nonlinear state-space representation of a recurrent HONU of second order yields, in vector form with respect to the state variable $\bar{\mathbf{x}}$

$$\begin{aligned} \bar{\mathbf{x}}(k+1) &= \bar{\mathbf{f}}(\bar{\mathbf{x}}(k)) + \bar{\mathbf{u}}(k) \\ \tilde{y}(k) &= \bar{x}_{n_y}(k). \end{aligned} \tag{61}$$

For development of the DHS method, let us express the respective incremental linear approximation via derivation of the Jacobian matrix of partial derivatives to the nonlinear state-space representation in (59) which yields the following

$$\mathrm{J}_m = \begin{bmatrix} \frac{\partial \bar{x}_1(k+1)}{\partial \bar{x}_1(k)} & \frac{\partial \bar{x}_1(k+1)}{\partial \bar{x}_2(k)} & \cdots & \cdots & \frac{\partial \bar{x}_1(k+1)}{\partial \bar{x}_{n_{\bar{x}}}(k)} \\ \vdots & \vdots & \ddots & & \vdots \\ \frac{\partial \bar{x}_{n_y}(k+1)}{\partial \bar{x}_1(k)} & \frac{\partial \bar{x}_{n_y}(k+1)}{\partial \bar{x}_2(k)} & \cdots & \frac{\partial \bar{x}_{n_y}(k+1)}{\partial \bar{x}_{n_{\bar{x}}-1}(k)} & \frac{\partial \bar{x}_{n_y}(k+1)}{\partial \bar{x}_{n_{\bar{x}}}(k)} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \frac{\partial \bar{x}_{n_{\bar{x}}}(k+1)}{\partial \bar{x}_1(k)} & \cdots & \cdots & \cdots & \frac{\partial \bar{x}_{n_{\bar{x}}}(k+1)}{\partial \bar{x}_{n_{\bar{x}}}(k)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{a}_{n_y,1} & \bar{a}_{n_y,2} & \bar{a}_{n_y,3} & \cdots & \bar{a}_{n_y,n_{\bar{x}}} \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{62}$$

Similarly, the Jacobian with respect to the input vector yields to be

$$\mathrm{J}_n = \begin{bmatrix} \frac{\partial \bar{x}_1(k+1)}{\partial \bar{u}(k)} & \frac{\partial \bar{x}_2(k+1)}{\partial \bar{u}(k)} & \cdots & \cdots & \frac{\partial x_{n_{\bar{x}}}(k+1)}{\partial \bar{u}(k)} \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 1 \end{bmatrix}^T. \tag{63}$$

Therefore, the previous state-space representation of (61) for the nonlinear HONU model of second order may be re-expressed in terms of the derived Jacobian matrices under similar notation to a linear HONU state-space model i.e.

$$\Delta\bar{\mathbf{x}}(k+1) = \bar{\mathbf{A}}(k) \cdot \Delta\bar{\mathbf{x}}(k) + \Delta\bar{\mathbf{u}}(k), \tag{64}$$

where $\Delta\bar{\mathbf{x}}(k)$ and $\Delta\bar{\mathbf{u}}(k)$ are apparent from (59) and $\bar{\mathbf{A}}(k)$ is the Jacobian matrix J_m of the recurrent HONU, indicated in (62), which maybe more explicitly stated as

$$\overline{\mathbf{A}}(k)=\frac{\partial\overline{\mathrm{f}}}{\partial\overline{\mathrm{x}}(k)}=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ : & ... & 1 & 0 & 0 \\ \overline{a}_{n_y,1} & \overline{a}_{n_y,2} & ... & ... & \overline{a}_{n_y,n_{\overline{x}}} \\ 0 & 0 & ... & 1 & 0 \\ 0 & 0 & ... & 0 & 0 \end{bmatrix}=\left[\{\overline{a}_{i,j}\}\right], \quad \begin{matrix} i=1...n_{\overline{x}}; \\ j=1...n_{\overline{x}} \end{matrix} \tag{65}$$

where the matrix $\overline{\mathbf{A}}(k)$ represents a $n_{\overline{x}}\times n_{\overline{x}}$ dimension matrix. For an arbitrary r order of HONU it yields that the matrix coefficients $\overline{\mathbf{A}}(k)$ may be expressed as

$$\overline{a}_{i,j}=\begin{cases} =1 & \text{for } i=2,3,...,n_x \wedge i\neq n_y;\ j=i+1 \\ =\boldsymbol{\psi}_j col^r(\mathbf{x}(k+1))=w[p]+\dfrac{\partial}{\partial\overline{x}_j}\left(\overline{x}_{n_y}(k)\right) & \text{for } j=1,2,...,n_{\overline{x}} \wedge i=n_y \\ =0 & \text{else} \quad , \end{cases} \tag{66}$$

where $col^r(\mathbf{x}(k+1))=\mathbf{x}(k+1)$ in the sense of a QNU, denotes the original neural weight vector $\mathbf{x}$ defined in (4) for one sample ahead. The term $w[n]$ denotes n^{th} element of $\mathbf{w}$ from the corresponding $\overline{x}_j(k)$ in vector $\mathbf{x}(k+1)$. Therefore, the newly introduced weight vector $\boldsymbol{\psi}$ for a QNU may be explicitly computed term-by-term as

$$\boldsymbol{\psi}_j=f(\mathbf{w})=\begin{cases} \displaystyle\sum_{l=0}^{n-1} w_{l,n} \\ \displaystyle\sum_{s=n}^{n_x} \alpha_{n,s}w_{n,s} \quad where \quad \alpha_{n,s}=1 \ \ \forall s\neq n \ \ \& \ \ \alpha_{n,n}=2. \end{cases} \tag{67}$$

For a practical example, refer to appendix section 13.1.

6.2 Pointwise State-Space Representation of Dynamic Neural Unit Models with Extension of HONU Feedback Control

Recalling the HONU structures in (8)-(9), its extension onto a HONU adaptive model yields a newly shaped neural model input vector where the variable u from equation (10) now acts as newly fed inputs to the engineering system. From this substitution, it can be seen that the vector of input variables u features previous output values of the engineering process or previous HONU model outputs in the sense of a dynamic HONU $\tilde{y}$, which must be re-expressed in terms of the respective HONU model state variables under the HONU matrix of dynamics $\overline{\mathbf{A}}(k)$. According to (10) the plant input with extension of a HONU feedback

controller now features a difference of the desired process set point value d and HONU feedback controller output q. Given this, the construction of state variable coefficients $\bar{a}_{i,j}$ of the matrix $\bar{\mathbf{A}}(k)$ starting with the first order LNU plant model with extended LNU feedback controller will be detailed.

6.2.1 Dynamic Linear Neural Models in Nonlinear Control Loop

If we consider a LNU model featuring an arbitrary length of previous model outputs n_y, and previous process inputs n_u, with the replacement of the control input according to (10) & (11), we may yield the following extended form

$$\begin{aligned}\tilde{y}(k) &= \sum_{i=0}^{n_x} w_i x_i = w_0 + w_1\tilde{y}(k-1) + w_2\tilde{y}(k-2) + \ldots + w_{n_y}\tilde{y}(k-n_y) + \\ & w_{n_y+1}[d(k-1) - p\cdot q(k-1)] + \ldots + w_{n_y+n_u}[d(k-n_u) - p\cdot q(k-n_u)].\end{aligned} \tag{68}$$

From (68) it is evident that to obtain the complete expansion of this HONU model, we must further express the previous HONU feedback controller outputs q, where as in the example of an LNU we may obtain

$$\begin{aligned} q(k-m) &= v_0 + \sum_{i=1}^{n_\xi} v_i \cdot \xi(k-m)_i \\ &for \quad m = 1,2,..,n_u. \end{aligned} \tag{69}$$

Thus following from expansion of relation (69) for $m = 1,2,..,n_u$ extending the definition of state variables with affine-control formulation of the input states, would yield the following state vector for arbitrary HONU feedback controllers

$$\bar{x}(k) = [\tilde{y}(k-n_y-1), \tilde{y}(k-5), \ldots, \tilde{y}(k), d(k-2n_y+1), \ldots, d(k)], \tag{70}$$

where the respective state variables $\bar{x}_i(k)$, $i = 1,2,..,(n_y + n_u + 2n_u = n_{\bar{x}})$ correspond to the step-delayed previous outputs of the HONU model with extension of a HONU feedback controller $\tilde{y}$, further step-delayed process inputs, now being the desired set point value d. Thus, the corresponding state equations yield to be as follows

$$
\begin{aligned}
&\overline{x}_1(k+1)=\overline{x}_2(k)\\
&\overline{x}_2(k+1)=\overline{x}_3(k)\\
&\quad\vdots\\
&\overline{x}_{n_y+n_u-1}(k+1)=\overline{x}_{n_y+n_u}(k)\\
&\overline{x}_{n_y+n_u}(k+1)=w_0+w_1\overline{x}_{n_y+n_u}(k)+w_2\overline{x}_{n_y+n_u-1}(k)+w_3\overline{x}_{n_y+n_u-2}(k)+\\
&w_{n_y+1}\overline{x}_{n_{\overline{x}}}(k)-w_{n_y+1}p\cdot q(k)+w_{n_y+2}\overline{x}_{n_{\overline{x}}-1}(k)-w_{n_y+2}p\cdot q(k-1)+\\
&w_{n_y+n_u}\overline{x}_{n_{\overline{x}}-2}(k)-w_{n_y+n_u}p\cdot q(k-2)\\
&\overline{x}_{n_y+n_u+1}(k+1)=\overline{x}_{n_y+n_u+2}(k)\\
&\quad\vdots\\
&\overline{x}_{n_{\overline{x}}-1}(k+1)=\overline{x}_{n_{\overline{x}}}(k)\\
&\overline{x}_{n_{\overline{x}}}(k+1)=\overline{d}(k),
\end{aligned}
\tag{71}
$$

where $q(k)=f(\overline{x}(k))$. Thus from the state equations represented in (71), a similar analogy maybe drawn to (64) for composition of the linear approximation of a LNU model with extension of a HONU feedback controller. The matrix $\overline{\mathbf{M}}(k)$ is now introduced, denoting the Jacobian matrix with feedback controller. $\overline{\mathbf{M}}(k)$ is now of $n_{\overline{x}}\times n_{\overline{x}}$ dimension where $n_{\overline{x}}=n_y+n_u+2n_u$ for an arbitrary length of LNU model

$$
\overline{\mathbf{M}}(k)=\frac{\partial\overline{\mathbf{g}}}{\partial\overline{\mathbf{x}}(k)}=\begin{bmatrix} 0 & 1 & 0 & 0 & 0\\ : & \dots & 1 & 0 & 0\\ \overline{\alpha}_{n_y+n_u,1} & \overline{\alpha}_{n_y+n_u,2} & \dots & \dots & \overline{\alpha}_{n_y+n_u,n_{\overline{x}}}\\ 0 & 0 & \dots & 1 & 0\\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}=\begin{bmatrix}\{\overline{\alpha}_{i,j}\}\\ i=1\dots n_{\overline{x}};\\ j=1\dots n_{\overline{x}}\end{bmatrix}.
\tag{72}
$$

Remark 1: We may state the following generalizations for an arbitrary length LNU model with extension of a HONU controller. The general vector of state variable may be defined as

$$
\begin{aligned}
\overline{\mathbf{x}}(k)&=\begin{bmatrix} x_1(k) & \dots & x_{n_y}(k) & x_{n_{y+1}}(k) & \dots & x_{n_{\overline{x}}}(k)\end{bmatrix}^T\\
&=\begin{bmatrix}\tilde{y}(k-n_y+1) & \dots & \tilde{y}(k) & d(k-n_u+1) & \dots & d(k)\end{bmatrix}^T.
\end{aligned}
\tag{73}
$$

Thus, with respect to definition of the new state variables as in (73), the following non-linear state space form may be defined for extension of the LNU with HONU feedback controller

$$
\overline{\mathbf{x}}(k+1)=\overline{\mathbf{g}}\left(\overline{\mathbf{x}}(k)\right)+\overline{\mathbf{d}}(k)\ ;\ \tilde{y}(k)=\overline{x}_{n_y}(k).
\tag{74}
$$

Thus, in similar analogy to (64) the linear approximation of the LNU model with extension of a HONU feedback controller may be defined as

$$\Delta\overline{\mathbf{x}}(k+1)=\overline{\mathbf{M}}(k)\cdot\Delta\overline{\mathbf{x}}(k)+\Delta\overline{\mathbf{d}}(k), \tag{75}$$

where the Jacobian matrix $\overline{\mathbf{M}}(k)$ with the feedback controller is defined by (72)

$$\overline{\mathbf{M}}(k)=\frac{\partial\overline{\mathbf{g}}}{\partial\overline{\mathbf{x}}(k)}=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ : & \dots & 1 & 0 & 0 \\ \overline{\alpha}_{n_y+n_u,1} & \overline{\alpha}_{n_y+n_u,2} & \dots & \dots & \overline{\alpha}_{n_y+n_u,n_{\overline{x}}} \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}=\left[\{\overline{\alpha}_{i,j}\}\right]_{\substack{i=1\dots n_{\overline{x}};\\ j=1\dots n_{\overline{x}}}}. \tag{76}$$

From the expanded forms (186)-(187) (appendix section 13.2.) it is thus apparent that the general form for an arbitrary length of LNU model with n_y previous output terms and n_u previous step-delayed process inputs it yields to be

$$\overline{\alpha}_{i,j}=\begin{cases} =1 & \text{for } i=2,3,\dots,n_{\overline{x}}\wedge i\neq n_y+n_u;\ j=i+1 \\ =\dfrac{\partial}{\partial\overline{x}_j}\left(\overline{x}_{n_{y+}n_u}(k)\right)=w[n]-\sum_{l=1}^{n_u}w_{ny+l}\cdot p\cdot v[n] & \text{for } j=1,2,\dots,n_{\overline{x}}\wedge i=n_y+n_u \\ =0 & \text{else}\quad . \end{cases} \tag{77}$$

The term $w[n]$ denotes n^{th} element of $\mathbf{w}$ from the corresponding $\overline{x}_j(k)$ in vector $\mathbf{x}(k+1)$. Analogically the n^{th} element of ξ is denoted by $v[n]$. If we consider the feedback controller to be a HONU of second order i.e. QNU then the explicit form for an arbitrary length is given by

$$\overline{\alpha}_{i,j}=\begin{cases} =1 & \text{for } i=2,3,\dots,n_{\overline{x}}\wedge i\neq n_y+n_u;\ j=i+1 \\ =\dfrac{\partial}{\partial\overline{x}_j}\left(\overline{x}_{n_{y+}n_u}(k)\right)=w[n]-\sum_{l=1}^{n_u}w_{ny+l}\cdot p\cdot v[n]\cdot col^r(\xi[n]) & \text{for } j=1,2,\dots,n_{\overline{x}}\wedge i=n_y+n_u \\ =0 & \text{else}\quad , \end{cases} \tag{78}$$

$$v[n]=\begin{cases} \sum_{l=0}^{n-1}v_{l,n} \\ \sum_{s=n}^{n_\xi}\alpha_{n,s}v_{n,s} \quad where \quad \alpha_{n,s}=1 \ \ \forall s\neq n \ \ \& \ \ \alpha_{n,n}=2, \end{cases} \tag{79}$$

where $w[n]$ denotes n^{th} element of $\mathbf{w}$ from the corresponding $\overline{x}_j(k)$ in vector $\mathbf{x}(k+1)$. similarly $col^r(\xi[n])$ corresponds to the n^{th} element from $col^r(\xi(k-l+1))$ multiplied by

$\overline{x}_j(k)$. For rigorous computation of the partial derivatives with respect to the state variable equations from (72) refer to the appendix section 13.2.

6.2.2 Dynamic Nonlinear Neural Models in Nonlinear Control Loop

In certain engineering processes, it can be more advantageous to apply nonlinear neural models for process identification. Typical cases maybe when batch training is required e.g. for process data influenced by noise on a nonlinear process or, use of a linear neural model may not be sufficient for identifying the nonlinear process characteristics as exampled in section 5, where a QNU was employed, further recent works include [33], [87] & [92]. To develop this method, let us consider restating a QNU via the following relation

$$\tilde{y}(k) = w_{0,0} + \sum_{i=1}^{n_x} w_{0,i} \cdot x_i + \sum_{i=1}^{n_x} \sum_{j=i}^{n_x} w_{i,j} \cdot x_i \cdot x_j, \tag{80}$$

where on substitution of the control law (10), the relation extends to

$$\begin{aligned} \tilde{y}(k) = w_{0,0} &+ \sum_{i=1}^{n_y} w_{0,i} \cdot x_i + \sum_{i=1}^{n_u} w_{0,n_y+i} \cdot u_i + \sum_{i=1}^{n_y} \sum_{j=i}^{n_y} w_{i,j} \cdot x_i \cdot x_j + \sum_{i=1}^{n_y} \sum_{j=i}^{n_u} w_{i,n_y+j} \cdot x_i \cdot u_j \\ &+ \sum_{i=1}^{n_u} \sum_{j=i}^{n_u} w_{ny+i,n_y+j} \cdot u_i \cdot u_j . \end{aligned} \tag{81}$$

The step delayed HONU controller outputs in the sense of a QNU for the control law may be restated as

$$\begin{aligned} q(k-m) &= v_{0,0} + \sum_{i=1}^{n_\xi} v_{0,i} \cdot \xi_i(k-m) + \sum_{i=1}^{n_\xi} \sum_{j=i}^{n_\xi} v_{i,j} \cdot \xi_i(k-m) \cdot \xi_j(k-m) \\ &for \quad m = 1,2,3, \end{aligned} \tag{82}$$

where m denotes the step-delayed samples of the HONU feedback controller outputs. Therefore in analogy, the matrix form (76) is also applicable. With this, the resulting general form for an arbitrary length of QNU model with n_y previous output terms and n_u previous step-delayed process inputs it yields to be

$$\overline{\alpha}_{i,j} = \begin{cases} = 1 & \text{for } i = 2,3,...,n_{\overline{x}} \wedge i \neq n_y + n_u \,;\; j = i+1 \\ = w[n] + \boldsymbol{\psi}_j col^r(\mathbf{x}(k+1)) + \dfrac{\partial}{\partial \overline{x}_j} \left(\displaystyle\sum_{l=1}^{n_u} \sum_{\kappa=l}^{n_u} w[l] \cdot u(k+1-l) \cdot u(k+1-\kappa) \right). & \\ \text{for } j = 1,2,...,n_{\overline{x}} \wedge i = n_y + n_u & \\ = 0 \quad \text{else} \quad . & \end{cases} \tag{83}$$

The general form for a QNU may be more explicitly defined as

$$\bar{\alpha}_{i,j}=\begin{cases}=1 \quad \text{for } i=2,3,...,n_{\bar{x}}\wedge i\neq n_y+n_u;\ j=i+1\\ =w[n]+\boldsymbol{\psi}_j col^r(\mathbf{x}(k+1))+\\ \left(\begin{array}{l}\sum_{l=0}^{n_y}\sum_{\kappa=0}^{n_{\sigma_q}} w_{l,n_y+1+\kappa}\frac{\partial}{\partial\bar{x}_j}(\xi(k+1)[l]\sigma_\kappa)+p\cdot\sum_{l=n_y+1}^{n_\xi}\sum_{\kappa=l}^{n_{\sigma_q}} w_{l,j}\frac{\partial}{\partial\bar{x}_j}(\xi(k+1)[l]\sigma_{\kappa-(n_y+1)}\\ +\sigma_{l-(n_y+1)}\xi(k+1)[\kappa])+p^2\sum_{l=0}^{n_{\sigma_q}}\sum_{\kappa=l}^{n_{\sigma_q}} w_{n_y+l,n_y+\kappa}\frac{\partial}{\partial\bar{x}_j}(\sigma_l\sigma_\kappa)\end{array}\right),\\ \text{for } j=1,2,...,n_{\bar{x}}\wedge i=n_y+n_u\\ =0 \quad \text{else} \quad ,\end{cases} \tag{84}$$

where σ denotes a vector of previous outputs as defined in (82) for one sample ahead, of the HONU feedback controller thus,

$$\sigma=[q(k),q(k-1),..,q(k-n_u+1)]. \tag{85}$$

Further, the general form for a CNU (HONU, r=3) then be defined as

$$\bar{\alpha}_{i,j}=\begin{cases}=1 \quad \text{for } i=2,3,...,n_{\bar{x}}\wedge i\neq n_y+n_u;\ j=i+1\\ =w[n]+\boldsymbol{\psi}_j col^r(\mathbf{x}(k+1))+\frac{\partial}{\partial\bar{x}_j}\left(\sum_{l=1}^{n_u}\sum_{m=l}^{n_u}\sum_{\kappa=m}^{n_u} w[l]\cdot u(k+1-l)\cdot u(k+1-m)\cdot u(k+1-\kappa)\right).\\ \text{for } j=1,2,...,n_{\bar{x}}\wedge i=n_y+n_u\\ =0 \quad \text{else} \quad .\end{cases} \tag{86}$$

From this section, a novel transformation of a HONU was derived to yield a state-space form of a classical HONU polynomial model and further, its extension with an additional HONU feedback controller for the composition of a HONU-MRAC control loop. Following the representations depicted in relations (65) and (76), motivation arises for utilizing the above representations as a means for analysing the stability of a HONU model and further its extensions with a HONU feedback controller, the next section thus aims to develop such condition.

6.3 BIBO Stability of Discrete-Time Pointwise State-Space HONU Representations (DHS)

From the previous section a novel representation of a HONU polynomial model into state-space representation was presented, with further extension as a HONU-MRAC closed control

loop. Based on the forms (65) and (76), in analogy to discrete time nonlinear model theory, several notions may be stated:

Definition 1: The evaluation of distance of (65) and (76) from their original HONU polynomial descriptions (1)-(3), may be evaluated via the following error margin ratio, where ε represents a small regularisation constant for zero value terms

$$\bar{\delta}(k) = e^{\left\| z\left(\bar{y}(k)\right) - z\left(\tilde{y}(k)\right) \right\|}, \tag{87}$$

where $\bar{y}(k)$ is the linearized output and $\tilde{y}(k)$ is the original nonlinear output and $z(.)$ is the z-score of the argument. Therefore, from relation (87), it is desirable that $\bar{\delta}(k) \doteq 1$ for high validity of BIBO stability conditions determined in (88), as the outputs of the HONU model or nonlinear control loop are closely resembled by the linearized approximation dynamics.

Theorem 1 (DHS): For LNUs the Jacobian matrices in (65) and (76) are time variant only when learning is on, while for higher orders the dynamics is variant already due to the variant Jacobian matrices in space. It may be stated that the HONU is BIBO stable and further asymptotically stable at a given state and actual input at sample *k* if

$$\rho(\bar{\mathbf{A}}(k)) < 1; \quad \rho(\bar{\mathbf{M}}(k)) < 1, \tag{88}$$

where $\rho(.)$ denotes the spectral radius of matrices $\bar{\mathbf{A}}(k)$ and $\bar{\mathbf{M}}(k)$ respectively. With the increasing order of HONU plant and further feedback controller $r, \gamma > 1,$ the respective Jacobian matrices are in fact time-variant without real time neural weight training.

Proof:

The statement (88) as applied to the pointwise state-space form (65) and (76), may be theoretically supported via means of the Lyapunov matrix condition. Let us select the following Lyapunov function candidate

$$V(\Delta\bar{\mathbf{x}}(k)) = \Delta\bar{\mathbf{x}}(k)^T \mathbf{P} \Delta\bar{\mathbf{x}}(k). \tag{89}$$

The Lyapunov function (89) is thus a positive definite function where for the state point $V(\Delta\bar{\mathbf{x}}(k)) = 0,$ further, for a non-zero equilibrium $\Delta\bar{\mathbf{x}}(k) \neq \Delta\bar{\mathbf{x}}_e$ about each fixed state we may state that the function $V(\Delta\bar{\mathbf{x}}(k)) = c$ holds as a lower bound for decrease of the function via an applied perturbation and $V(\Delta\bar{\mathbf{x}}(k)) > c$ for all other values $\neq x_e$. Therefore, in order to assess the asymptotic stability of (89) we may assess the Lyapunov function derivative where

$$\begin{aligned}&\Delta V(\Delta\overline{\mathbf{x}}(k)) \tilde{=} V(\Delta\overline{\mathbf{x}}(k)) - V(\Delta\overline{\mathbf{x}}(k-1)) < 0, \forall \Delta\overline{\mathbf{x}}(k) \in \mathbb{R}\\&V(\Delta\overline{\mathbf{x}}(k)) \to \infty.\end{aligned} \tag{90}$$

Then, two following statements are thus equally justified for asymptotic stability [85].

1. The time-invariant HONU matrix $\overline{\mathbf{A}}(k)$ is asymptotically stable if and only if the relation (88) holds.
2. Given any matrix $\mathbf{Q} = \mathbf{Q}^T > 0$ there exists a positive definite matrix $\mathbf{P} = \mathbf{P}^T$ such that satisfies the proceeding relation

$$\overline{\mathbf{A}}(k)^{\ T}\mathbf{P}\overline{\mathbf{A}}(k) - \mathbf{P} = -\mathbf{Q}. \tag{91}$$

Therefore, we may consider the linearized switched discrete time representation of the HONU as

$$\begin{aligned}\Delta\overline{\mathbf{x}}(k) &= \overline{\mathbf{A}}(k)\Delta\overline{\mathbf{x}}(k-1) + \Delta\overline{\mathbf{u}}(k),\\\Delta\overline{\mathbf{x}}(k-1) &= \overline{\mathbf{A}}(k)\Delta\overline{\mathbf{x}}(k-2) + \Delta\overline{\mathbf{u}}(k),\end{aligned} \tag{92}$$

where $\Delta\overline{\mathbf{u}}(k)$ is considered as a positive, constant input within neighbourhood $\Delta\overline{\mathbf{x}}_e$. From (92), it yields that

$$\overline{\mathbf{A}}(k)^{-1}\Delta\overline{\mathbf{x}}(k-1) - \overline{\mathbf{A}}(k)^{-1}\Delta\overline{\mathbf{u}}(k) = \Delta\overline{\mathbf{x}}(k-2). \tag{93}$$

Substituting the above Lyapunov function candidate into the expression (90) yields the following

$$\therefore V(\Delta\overline{\mathbf{x}}(k)) - V(\Delta\overline{\mathbf{x}}(k-1)) = \Delta\overline{\mathbf{x}}(k)^T\mathbf{P}\Delta\overline{\mathbf{x}}(k) - \Delta\overline{\mathbf{x}}(k-1)^T\mathbf{P}\Delta\overline{\mathbf{x}}(k-1). \tag{94}$$

Therefore, the following expression may be obtained

$$\begin{aligned}&\Delta\overline{\mathbf{x}}(k-1)^T\overline{\mathbf{A}}(k)^{\ T}\mathbf{P}\overline{\mathbf{A}}(k)\Delta\overline{\mathbf{x}}(k-1) - \Delta\overline{\mathbf{x}}(k-1)^T\mathbf{P}\Delta\overline{\mathbf{x}}(k-1)\\&= \Delta\overline{\mathbf{x}}(k-1)^T(\overline{\mathbf{A}}(k)^{\ T}\mathbf{P}\overline{\mathbf{A}}(k) - \mathbf{P})\Delta\overline{\mathbf{x}}(k-1).\end{aligned} \tag{95}$$

However according to Lyapunov's matrix equation in (91), the final sequence may be obtained

$$\begin{aligned}&\Delta\overline{\mathbf{x}}(k-1)^T(\overline{\mathbf{A}}(k)^{\ T}\mathbf{P}\overline{\mathbf{A}}(k) - \mathbf{P})\Delta\overline{\mathbf{x}}(k-1) \Rightarrow -\sum_{k=0}^{N-1}\Delta\overline{\mathbf{x}}(k-1)^T\mathbf{Q}\Delta\overline{\mathbf{x}}(k-1)\\&where\ N \to \infty.\end{aligned} \tag{96}$$

Which is in fact strictly decreasing as $\mathbf{Q}$ is a positive matrix, where $\mathbf{Q} = \mathbf{Q}^T > 0$. Therefore, for any values $\Delta\overline{\mathbf{x}}(k) \neq \Delta\overline{\mathbf{x}}_e$ in neighbourhood of the given state, the Lyapunov function can only decrease. However, as the Lyapunov function is bounded from below by $V(\Delta\overline{\mathbf{x}}(k)) = c$ it

must stop decreasing at the lower bound, which directly means the local asymptotic stability of the system defined in (92). Furthermore, that if (92) is asymptotically stable it is only possible given that (88) holds in each given state point.

Remark 2: As (92) is a the linearized switched discrete time system of a HONU model or closed control loop a more global condition can be stated for the set of matrices i.e.

$$\bar{\mathbf{A}}_m = \{\bar{\mathbf{A}}(k-k_0), \bar{\mathbf{A}}(k-k_0+1), .. \bar{\mathbf{A}}(k)\}, \tag{97}$$

where the spectral radii $\rho(.)$ computed for the set (97) yields the joint spectral radii of the linearized switched discrete time system (65) and further (76). Thus, the global asymptotic stability & BIBO stability across the set $\bar{\mathbf{A}}_m$ (97) is justified if

$$\rho(\bar{\mathbf{A}}_m) = \rho\left(\left[\bar{\mathbf{A}}(k-k_0) .. \bar{\mathbf{A}}(k) \right]^{\frac{1}{k}} \right) < 1; \rho(\bar{\mathbf{M}}_m) < 1. \tag{98}$$

The closed loop variation of (92) and (98) where, the matrix $\bar{\mathbf{M}}(k)$ and input term $\Delta\bar{\mathbf{d}}(k)$ are used may be analogically justified.

7 DDHS: Decomposed Discrete-Time Higher Order Neural Unit Stability

In the preceding chapter of this dissertation, a transformation was derived to restructure a HONU in the form of relations (1)-(2) to a linearized switched discrete time form denoted in relations (65) and (76). Given, this the DHS condition was composed as a practical means of assessing the BIBO further, asymptotical stability of linearized representation of HONU models and their extension as a closed HONU-MRAC control loop.

Although a great discipline of industrial systems can be identified and controlled with $r;\gamma \leq 2$ polynomial order, the use of higher order control loops as exampled in [87] may be limited due to the high error margin ratio about the localized operating state point due to linearization. Therefore, an enhancement can be found in the very fundamental structure of HONUs. This section thus presents the decomposed discrete time higher order neural unit (DDHS) approach and stronger ISS stability condition for justification of BIBS stability of HONUs and their nonlinear control loop.

7.1 Decomposed Pointwise State-Space Representation of Dynamic Nonlinear Neural Unit Models

To develop this approach, let us consider the expanded form of a QNU i.e. HONU, $r=2$ for the input vector (4), where an arbitrary length of previous model outputs n_y and previous process inputs n_u are considered. Then, on observation of the expanded form (50) we may restate the QNU with respect to the principle input vector $\mathbf{x}(k-1)$ as

$$\tilde{y}(k)=\sum_{i=0}^{n_x}\sum_{j=i}^{n_x}w_{i,j}\cdot x_i\cdot x_j=w_{0,0}+\sum_{i=1}^{n_y}x_i\cdot\left(w_{0,i}+\sum_{j=i}^{n_x}w_{i,j}\cdot x_j\right)+\sum_{i=n_y+1}^{n_x}x_i\cdot\left(w_{0,i}+\sum_{j=i}^{n_x}w_{i,j}\cdot x_j\right), \tag{99}$$

where $n_x=n_y+n_u$. Then, similarly to the definition (4) we may redefine a new set of vectors as

$$\hat{\mathbf{x}}(k-1)=[\tilde{y}(k-n_y)\quad \tilde{y}(k-n_y+1)\quad ..\quad \tilde{y}(k-1)]^T, \tag{100}$$

$$\hat{\mathbf{u}}(k-1)=\left[u(k-n_u)\quad u(k-n_u+1)\quad ...\quad u(k-1)\right]^T. \tag{101}$$

Thus the definition (2) maybe restated as

$$\tilde{y}(k)=\mathbf{w}(k)\cdot col^r\left(\mathbf{x}(k\text{-}1)\right);\mathbf{x}(k-1)=\begin{bmatrix}1\\ \hat{\mathbf{x}}(k-1)\\ \hat{\mathbf{u}}(k-1)\end{bmatrix}. \tag{102}$$

Further, the form (99) when considering the relations (100)-(102) may be summarized as

$$\hat{x}_{n_y}(k)=w_{0,0}+\sum_{i=1}^{n_y}\hat{x}_i(k-1)\cdot\hat{a}_i+\sum_{i=1}^{n_u}\hat{u}_i(k-1)\cdot\hat{b}_i, \tag{103}$$

where the coefficients $\hat{a}_i$ and $\hat{b}_i$ maybe respectively defined in sub polynomial form as in the sense of a QNU as

$$\hat{a}_i=\hat{a}_i\left(\hat{\mathbf{x}}(k-1),\hat{\mathbf{u}}(k-1),\mathbf{w}\right)=w_{0,i}+\sum_{j=i}^{n_x}w_{i,j}\cdot x_j(k-1), \tag{104}$$

$$\hat{b}_i=\hat{b}_i\left(\hat{\mathbf{u}}(k-1),\mathbf{w}\right)=w_{0,i}+\sum_{j=i}^{n_x}w_{i,j}\cdot x_j(k-1)\ ;\ i>n_y. \tag{105}$$

For the fully derived forms of (104)-(105) refer to the appendix section 13.3. For the explicit forms for further orders, refer to the Table 5. Thus, given the state variable vector definition

as per (100), the original HONU polynomial form may be re-expressed in canonical state-space form as

$$\begin{aligned}\hat{\mathbf{x}}(k) &= \hat{\mathbf{A}}(k\text{-}1)\cdot\hat{\mathbf{x}}(k-1)+\hat{\mathbf{B}}(k-1)\cdot\hat{\mathbf{u}}(k-1)+\hat{\mathbf{w}}_0(k\text{-}1)\\ \tilde{y}(k) &= \hat{\mathbf{C}}\cdot\hat{\mathbf{x}}(k)\ ;\ \hat{\mathbf{w}}_0=\begin{bmatrix}0 & \dots & 0 & w_{0,0}\end{bmatrix}^T,\end{aligned} \tag{106}$$

where the matrix of dynamics $\hat{\mathbf{A}}(k\text{-}1)$ and input matrix $\hat{\mathbf{B}}(k-1)$ respectively may be defined as

$$\hat{\mathbf{A}}=\begin{bmatrix}0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & \vdots\\ 0 & 0 & \ddots & \ddots & 0\\ 0 & 0 & \ddots & 0 & 1\\ \hat{a}_{n_y} & \hat{a}_{n_y-1} & \cdots & \hat{a}_2 & \hat{a}_1\end{bmatrix},\ \hat{\mathbf{B}}=\begin{bmatrix}0 & 0 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 0\\ \hat{b}_{n_u} & \hat{b}_{n_u-1} & \cdots & \hat{b}_1\end{bmatrix}. \tag{107}$$

Further, the output matrix $\hat{\mathbf{C}}=\begin{bmatrix}0 & \cdots & 0 & 1\end{bmatrix}^T$. However, we may further simplify this expression on considering the neural weight bias as an additional input vector $w_{0,0}=w_{0,0}(k)$. Therefore, the augmented input matrix and input vector may be defined as

$$\hat{\mathbf{x}}(k)=\hat{\mathbf{A}}(k\text{-}1)\cdot\hat{\mathbf{x}}(k-1)+\hat{\mathbf{B}}_{\mathbf{a}}\cdot\hat{\mathbf{u}}_{\mathbf{a}}(k-1)\ ;\ \tilde{y}(k)=\hat{\mathbf{C}}\cdot\hat{\mathbf{x}}(k-1), \tag{108}$$

where

$$\hat{\mathbf{B}}_{\mathbf{a}}=\begin{bmatrix}\hat{\mathbf{B}} & \begin{matrix}0\\ \vdots\\ 0\\ 1\end{matrix}\end{bmatrix},\quad \hat{\mathbf{u}}_{\mathbf{a}}(k\text{-}1)=\begin{bmatrix}\hat{\mathbf{u}}(k\text{-}1) & w_{0,0}(k\text{-}1)\end{bmatrix}^T. \tag{109}$$

$\hat{\mathbf{A}}$ results in a time-variant matrix of dynamics for polynomial orders r>1 and it is also further due to applied learning; thus, the matrix $\hat{\mathbf{A}}$ will further be referenced here on as the Local Matrix of Dynamics (LMD).

TABLE 5

EXPLICIT FORMS OF MATRIX COEFFICIENTS FOR HONU MODELS

Index Range i	**General form of** $\hat{a}_i$, **and** $\hat{b}_i$ **coefficients**
QNU (HONU, r**=2)**	
$1\dots n_y$	$\hat{a}_i = w_{0,i} + \sum_{j=i}^{n_y} w_{i,j}\tilde{y}(k-j) + \sum_{j=1}^{n_u} w_{i,j+n_y}u(k-j)$
$1\dots n_u$	$\hat{b}_i = w_{0,n_y+i} + \sum_{j=i}^{n_u} w_{n_y+i,n_y+j}u(k-j)$

CNU (HONU, r=3)	
$1\ldots n_y$	$\hat{a}_i = w_{0,0,i} + \left[\sum_{l=1}^{n_y} w_{0,i,l} \tilde{y}(k-l) + \sum_{l=1}^{n_u} w_{0,i,l+n_y} u(k-l) \right]$ $+ \sum_{j=i}^{n_x} x_j \left[\sum_{l=j}^{n_y} w_{i,j,l} \tilde{y}(k-l) + \sum_{l=j}^{n_x} w_{i,j,l} u(k-(l-n_y)) \right]$
$1\ldots n_u$	$\hat{b}_i = w_{0,0,n_y+i} + \sum_{l=1}^{n_u} w_{0,n_y+i,n_y+l} u(k-l) +$ $\sum_{j=1}^{n_u} u(k-j) \cdot \sum_{l=j}^{n_u} w_{n_y+i,n_y+j,n_y+l} u(k-l)$

7.2 Decomposed Pointwise State-Space Representation of Dynamic Neural Unit Models with Extension of HONU Feedback Control

Recalling the HONU structures denoted in (1)-(2) with respect to the adaptive HONU feedback controller, its extension onto a previously identified HONU adaptive model yields a newly shaped neural model input vector where the control law u from equation (10)-(11) now acts as newly fed inputs to the engineering system. From this substitution, it can also be seen that the vector of input variables $\hat{\mathbf{u}}$ (101) yield previous output values of the dynamic HONU model $\tilde{y}$, which must be re-expressed in terms of the respective HONU model state variables under the LMD $\hat{\mathbf{A}}$.This section aims to derive the decomposition of HONU models with extension of their control loop as a canonical state-space form for $r;\gamma>1$ polynomial order.

7.2.1 Dynamic Linear Neural Models as Nonlinear Control Loop

LNUs have proven their advantages as computationally efficient and effective approximators, further optimizers of linear and low to moderately non-linear dynamic systems, this section aims to present a novel state-space representation of such architecture which was most recently presented in the work [89]. To develop this approach let us consider, the expanded form of a LNU from (1) with the control law (10)-(11) as follows

$$\begin{aligned} &\tilde{y}(k) = w_0 + w_1 \tilde{y}(k-1) + w_2 \tilde{y}(k-2) + .. + w_{n_y} \tilde{y}(k-n_y) + \\ &w_{n_{y+1}} [d(k-1) - p \cdot q(k-1)] + .. + w_{n_y+n_u} [d(k-n_u) - p \cdot q(k-n_u)], \end{aligned} \tag{110}$$

upon expansion of the terms in (110) we may restate that

$$\tilde{y}(k) = w_0 + \sum_{j=i}^{n_y} w_i \hat{x}(k-j) + \sum_{j=1}^{n_u} w_{j+n_y} \hat{u}(k-j) - p \cdot \sum_{j=1}^{n_u} w_{n_y+j} \cdot q(\mathbf{v}, \hat{x}(k-j), \hat{u}(k-j)), \tag{111}$$

Due to the extension of a HONU feedback controller, we may introduce the extended vector of state variables as

$$\hat{\mathbf{x}}(k\text{-}1) = [\tilde{y}(k-(n_y+n_u)) \quad .. \quad \tilde{y}(k\text{-}2) \quad \tilde{y}(k\text{-}1)]^T. \tag{112}$$

Further, the input vector for the whole non-linear control loop may be defined as

$$\hat{\mathbf{u}}(k\text{-}1) = [d(k-(n_u+n_u)) \quad .. \quad d(k\text{-}2) \quad d(k\text{-}1)]^T. \tag{113}$$

Then, due to the final term in (111) containing previous step delayed output terms $\tilde{y}(k\text{-}1)$ from the previous sample vector $\hat{\mathbf{x}}(k\text{-}1)$ in (112), further previous step-delayed input terms from the previous sample vector $\hat{\mathbf{u}}(k\text{-}1)$ in (113) the form (111) may be re-expressed as follows

$$\tilde{y}(k) = \sum_{i=1}^{n_y+n_u} \hat{x}_i(k-i) \cdot \hat{\alpha}_i + \sum_{i=1}^{n_u+n_u} \hat{u}_i(k-i) \hat{\beta}_i + C_i(\mathbf{w}_0). \tag{114}$$

For simplification, the operator $C_i(.)$ is now introduced to denote the sum of constant neural bias weight terms. Then the local characteristic coefficients for an LNU with extension of a HONU feedback controller of arbitrary length may be expressed as

$$\hat{\alpha}_i = \begin{cases} w_i - p \cdot \sum_{j=1}^{n_u} w_{n_y+j} \cdot C_i(q(k-j)) \quad for \quad i = 1, 2, 3, .., n_y \\ -p \cdot \sum_{j=1}^{n_u} w_{n_y+j} \cdot C_i(q(k-j)) \quad for \quad i = n_y+1, .., n_y+n_u, \end{cases} \tag{115}$$

where $C_i(.)$ here denotes the sum of coefficients for the $\hat{x}_i(k\text{-}1)$ element. Further, for the coefficients concerning the input vector it yields that

$$\hat{\beta}_i = \begin{cases} w_{n_y+i} - p \cdot \sum_{j=1}^{n_u} w_{n_y+j} \cdot C_i(q(k-j)) \quad for \quad i = 1, .., n_u \\ -p \cdot \sum_{j=1}^{n_u} w_{n_y+j} \cdot C_i(q(k-j)) \quad for \quad i = n_u+1, .., n_u+n_u. \end{cases} \tag{116}$$

As a practical example, section 13.4 details the derivation for a LNU with extension of a QNU feedback controller for $n_y = 4$ and $n_u = 3$ step-delayed samples of the principle input

vector $\mathbf{x}$. Given the definitions (115)-(116), then for a closed loop LNU with nonlinear control loop it leads to the following canonical state-space form

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{M}}(k\text{-}1)\cdot\hat{\mathbf{x}}(k-1) + \hat{\mathbf{N}}_{\mathbf{a}}\cdot\hat{\mathbf{u}}_{\mathbf{a}}(k-1) \;\; ; \;\; \tilde{y}(k) = \hat{\mathbf{C}}\cdot\hat{\mathbf{x}}(k), \tag{117}$$

where

$$\hat{\mathbf{M}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & 0 & 1 \\ \hat{\alpha}_{n_{y+}n_u} & \hat{\alpha}_{n_{y+}n_u-1} & \cdots & \hat{\alpha}_2 & \hat{\alpha}_1 \end{bmatrix}, \; \hat{\mathbf{N}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \hat{\beta}_{n_u+n_u} & \hat{\beta}_{(n_u+n_u)-1} & \cdots & \hat{\beta}_1 \end{bmatrix}. \tag{118}$$

Further, that augmented input matrix and input vector may be defined as

$$\hat{\mathbf{N}}_{\mathbf{a}} = \begin{bmatrix} & 0 \\ \hat{\mathbf{N}} & \vdots \\ & 0 \\ & 1 \end{bmatrix}, \quad \hat{\mathbf{u}}_{\mathbf{a}}(k\text{-}1) = \begin{bmatrix} \hat{\mathbf{u}}(k\text{-}1) & C_i(\mathbf{w}_0) \end{bmatrix}^T. \tag{119}$$

The matrix $\hat{\mathbf{M}}$ represents a $n_{\hat{x}} \times n_{\hat{x}}$ matrix where $n_{\hat{x}} = n_y + n_u$. The row index corresponding to the position $n_{\hat{x}}$ yields a set of coefficients $\hat{\alpha}_i$ Similarly, the matrix $\hat{\mathbf{N}}$ yields to be of a dimension $n_{\hat{x}} \times 2n_u$ representing the state input matrix corresponding to the state variables of HONU vectors $\mathbf{x}$ and ξ.

7.2.2 Dynamic Nonlinear Neural Models as Nonlinear Control Loop

As recent works [33], [87] & [92] have shown, nonlinear HONU models and their closed control loop are advantageous for application to nonlinear dynamic processes or linear oscillating systems, as well as optimization of controller convergence for linear systems. Due to the complexity in analytical description, let us consider at first a purely state feedback controller, i.e. let's define the controller input vector as

$$\boldsymbol{\xi}(k) = [\xi_0 = 1 \quad \xi_2 \quad .. \quad \xi_{n_y}]^T = \begin{bmatrix} 1 \\ \hat{\mathbf{x}}(k) \end{bmatrix}. \tag{120}$$

On considering the control law (10)-(11) with the controller input vector (120), the expanded form of a QNU with HONU feedback controller yields

$$\tilde{y}(k)=w_{0,0}+\sum_{i=1}^{n_y}w_{0,i}\cdot x_i+\sum_{i=1}^{n_u}w_{0,n_y+i}\cdot\left[d_i-p\cdot v_i\cdot\hat{x}_i\right]+\sum_{i=1}^{n_y}\sum_{j=i}^{n_y}w_{i,j}\cdot x_i\cdot x_j+$$
$$\sum_{i=1}^{n_y}\sum_{j=i}^{n_u}w_{i,n_y+j}\cdot x_i\cdot\left[d_j-p\cdot v_j\cdot\hat{x}_j\right]+\sum_{i=1}^{n_u}\sum_{j=i}^{n_u}w_{ny+i,n_y+j}\cdot\left[d_i-p\cdot v_i\cdot\hat{x}_i\right]\cdot\left[d_j-p\cdot v_j\cdot\hat{x}_j\right]. \tag{121}$$

Then, as terms $\hat{x}$ are purely step-delayed we may express the canonical state-space form for a QNU, with CNU controller for the controller input (120) as

$$\hat{\mathbf{x}}(k)=\left[\hat{\mathbf{A}}-\hat{\mathbf{B}}\cdot\hat{\mathbf{Q}}\right]\cdot\hat{\mathbf{x}}(k-1)+\hat{\mathbf{N}}_{\mathbf{a}}\cdot\hat{\mathbf{u}}_{\mathbf{a}}(k-1)\ ;\ \tilde{y}(k)=\hat{\mathbf{C}}\cdot\hat{\mathbf{x}}(k), \tag{122}$$

where considering the adaptive feedback gain and controller weights are adaptive, it yields

$$\hat{\mathbf{Q}}=\begin{bmatrix} p(k-n_y)\cdot\left[v_{0,0,1}+\sum_{j=1}^{n_y}\sum_{\kappa=j}^{n_y}v_{1,j,\kappa}\cdot x_j\cdot x_k \quad v_{0,0,2}+\sum_{j=2}^{n_y}\sum_{\kappa=j}^{n_y}v_{2,j,\kappa}\cdot x_j\cdot x_k \quad\cdots\quad v_{0,0,n_y}+v_{n_y,n_y,n_y}\cdot x_{n_y}{}^2\right] \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ p(k-2)\cdot\left[v_{0,0,1}+\sum_{j=1}^{n_y}\sum_{\kappa=j}^{n_y}v_{1,j,\kappa}\cdot x_j\cdot x_k \quad v_{0,0,2}+\sum_{j=2}^{n_y}\sum_{\kappa=j}^{n_y}v_{2,j,\kappa}\cdot x_j\cdot x_k \quad\cdots\quad v_{0,0,n_y}+v_{n_y,n_y,n_y}\cdot x_{n_y}{}^2\right] \\ p(k-1)\cdot\left[v_{0,0,1}+\sum_{j=1}^{n_y}\sum_{\kappa=j}^{n_y}v_{1,j,\kappa}\cdot x_j\cdot x_k \quad v_{0,0,2}+\sum_{j=2}^{n_y}\sum_{\kappa=j}^{n_y}v_{2,j,\kappa}\cdot x_j\cdot x_k \quad\cdots\quad v_{0,0,n_y}+v_{n_y,n_y,n_y}\cdot x_{n_y}{}^2\right] \end{bmatrix}. \tag{123}$$

Further, the augmented input matrix $\hat{\mathbf{N}}_{\mathbf{a}}$ and augmented input vector $\hat{\mathbf{u}}_{\mathbf{a}}$ for the closed loop may be defined as follows

$$\hat{\mathbf{N}}_{\mathbf{a}}=\begin{bmatrix}\mathbf{I} & \begin{matrix}0\\ \vdots\\ 0\\ 1\end{matrix}\end{bmatrix},\qquad \hat{\mathbf{u}}_{\mathbf{a}}=\begin{bmatrix} d(k-n_y)-p(k-n_y)\cdot v_{0,0,0}(k-n_y)\\ \vdots\\ d(k-2)-p(k-2)\cdot v_{0,0,0}(k-2)\\ d(k-1)-p(k-1)\cdot v_{0,0,0}(k-1)\\ w_{0,0}\end{bmatrix}, \tag{124}$$

where $\hat{\mathbf{N}}_{\mathbf{a}}$ is a $n_{\hat{x}}\times(n_{\hat{x}}+1)$ matrix, $\mathbf{I}$ is $n_{\hat{x}}\times n_{\hat{x}}$ identity matrix, and neural bias weights of a controller, i.e. $v_{0,0,0}$ are considered as time-varying inputs for this nonlinear state space decomposition.

Remark 3: *HONU with Extension of Feedback controller with Inputs*

In practical applications, it can be necessary to further incorporate previous step-delayed inputs i.e. the desired value d in the sense of the closed HONU-MRAC control loop. In such

case, the definitions (117)-(119) are also valid. The explicit numerical forms for computation of the local characteristic coefficients $\hat{\alpha}_i$ and $\hat{\beta}_i$ for HONU models with extension of HONU controllers i.e. $r;\gamma \geq 2$ are therefore defined in the Table 6.

TABLE 6

EXPLICIT FORMS OF MATRIX COEFFICIENTS FOR HONU-MRAC LOOP

Index Range i	**General form of** $\hat{\alpha}_i$ **and** $\hat{\beta}_i$ **coefficients**
QNU (Plant) +HONU (Feedback Controller)	
$1\ldots\left(n_y+n_u\right)$	$\hat{\alpha}_i = w_{0,i} + \sum_{j=i}^{n_y} w_{i,j}\tilde{y}(k-j) + \sum_{j=1}^{n_u} w_{i,j+n_y}[d(k-j)-p\cdot q(k-j)]$ $+\sum_{j=1}^{n_u} w_{0,n_y+j}\cdot C_i(-p\cdot q(k-j)) + \sum_{l=1}^{n_u}\sum_{j=l}^{n_u} w_{n_y+l,n_y+j} C_m([d(k-l)-p\cdot q(k-l)]\cdot$ $[d(k-j)-p\cdot q(k-j)])$
$1\ldots 2n_u$	$\hat{\beta}_i = w_{0,n_y+i} + \sum_{l=1}^{n_u}\sum_{j=l}^{n_u} w_{n_y+l,n_y+j}\left[C_i(d(k-j)) + C_i(-p\cdot q(k-j))\right.$ $\left.+C_i(-p\cdot q(k-l)d(k-j)) + C_i(p^2\cdot q(k-l)q(k-j))\right]\cdot$
CNU (Plant) +HONU (Feedback Controller)	
$1\ldots\left(n_y+n_u\right)$	$\hat{\alpha}_i = w_{0,0,i} + \left[\sum_{l=i}^{n_y} w_{0,i,l}\tilde{y}(k-l) + \sum_{l=1}^{n_u} w_{0,i,l+i}[d(k-l)-p\cdot q(k-l)]\right]$ $+\sum_{j=i}^{n_x} x_j\left[\sum_{l=i}^{n_y} w_{i,j,l}\tilde{y}(k-l) + \sum_{l=j}^{n_x} w_{i,j,l}[d(k-(l-n_y))-p\cdot q(k-(l-n_y))]\right]+$ $\sum_{l=1}^{n_u} w_{0,0,l}\cdot C_i([d(k-l)-p\cdot q(k-l)])+$ $\left[\sum_{l=1}^{n_u}\sum_{j=l}^{n_u} w_{0,n_y+l,j} C_i([d(k-l)-p\cdot q(k-l)]\cdot[d(k-j)-p\cdot q(k-j)])\right]$ $+\left[\sum_{m=1}^{n_u}\sum_{j=m}^{n_u}\sum_{l=j}^{n_u} w_{n_y+m,n_y+j,n_y+l} C_i([d(k-m)-p\cdot q(k-m)]\cdot[d(k-j)-p\cdot q(k-j)\right.$ $\left.\cdot[d(k-l)-p\cdot q(k-l)])\right]$

$1\ldots 2n_u$	$\hat{\beta}_i = w_{0,0,n_y+i} + \sum_{l=1}^{n_u}\sum_{j=l}^{n_u} w_{n_y+l,n_y+j}\left[C_i(d(k-j)) + C_i(-p\cdot q(k-j))\right.$ $+C_i(-p\cdot q(k-l)d(k-j)) + C_i(p^2\cdot q(k-l)q(k-j))] +$ $\left[\sum_{m=1}^{n_u}\sum_{j=m}^{n_u}\sum_{l=j}^{n_u} w_{n_y+m,n_y+j,n_y+l} C_i([d(k-m)-p\cdot q(k-m)]\cdot[d(k-j)-p\cdot q(k-j)]\right.$ $\left.\cdot[d(k-l)-p\cdot q(k-l)])\right]$

7.3 An ISS Approach to BIBS Stability of Decomposed Pointwise State-Space HONU Representations (DDHS)

From the previous subsections, an approach via re-expression of a HONU model with extension of a HONU feedback controller, into a decomposition of sub polynomials was derived. To conclude this chapter, the main result of this dissertation is the DDHS method for stability evaluation & monitoring of the discrete time adaptive control loop in real time. From the principles of input-to-state stability (ISS) [45], [76] and definitions of BIBO and BIBS stability [48], [85] the following may be stated:

Definition 2: From [45] the decomposed state-space representation (108) is ISS stable provided that

$$\left\|\overline{\mathbf{x}}(k)\right\| \le \beta(\left\|\overline{\mathbf{x}}(k_0)\right\|) + \gamma(\left\|u(k)\right\|_\infty), \tag{125}$$

where $\beta(\cdot)$ represents a function of class κL which is asymptotically stable such that the function converges to a minimum for $k \to \infty$. Moreover, for a zero equilibrium that $\beta(\cdot) \to 0$. Further, $\gamma(\cdot)$ represents a class κ_∞ function which is unbounded and strictly increasing such that about an initial zero state that $\gamma = 0$ and for $k \to \infty, \gamma \to \infty$. For derivation of the DDHS method, let us firstly consider the general solution of a discrete time nonlinear state space system and its expression in state space as

$$\mathbf{x}(k) = \prod_{\kappa=k_0}^{k-1} \mathbf{A}(\kappa)\mathbf{x}(k_0) + \sum_{\kappa=k_0}^{k-1}\prod_{i=\kappa}^{k-1} \mathbf{A}(i)\cdot\mathbf{B}(\kappa)\mathbf{u}(\kappa), \tag{126}$$

where,

$$\mathbf{x}(k+1) = \mathbf{A}(k)\left[\prod_{\kappa=k_0}^{k-1} \mathbf{A}(\kappa)\mathbf{x}(k_0) + \sum_{\kappa=k_0}^{k-1}\prod_{i=\kappa}^{k-1} \mathbf{A}(i)\cdot\mathbf{B}(\kappa)\mathbf{u}(\kappa)\right] + \mathbf{B}(k)\mathbf{u}(k). \tag{127}$$

Then considering the definition 2, we can take the norm of both sides of (126) which yields

$$\begin{aligned}\|\mathbf{x}(k)\| &\leq \left\|\prod_{\kappa=k_0}^{k-1}\mathbf{A}(\kappa)\mathbf{x}(k_0)\right\| + \left\|\sum_{\kappa=k_0}^{k-1}\prod_{i=\kappa}^{k-1}\mathbf{A}(i)\cdot\mathbf{B}(\kappa)\mathbf{u}(\kappa)\right\|, \\ \|\mathbf{x}(k)\| &\leq \left\|\prod_{\kappa=k_0}^{k-1}\mathbf{A}(\kappa)\right\|\cdot\|\mathbf{x}(k_0)\| + \sum_{\kappa=k_0}^{k-1}\left\|\prod_{i=\kappa}^{k-1}\mathbf{A}(i)\cdot\mathbf{B}(\kappa)\mathbf{u}(\kappa)\right\|.\end{aligned} \tag{128}$$

Then, considering the solution as a summation of homogenous and particular solutions, we may via the adoption of the BIBS discrete-time variant systems theory of [85] setting $\hat{\mathbf{u}}_{\mathbf{a}}(\kappa)=0$ then for all $\forall k \neq k_0$ if the following holds

$$\|\mathbf{x}(k)\| \leq \left\|\prod_{\kappa=k_0}^{k-1}\mathbf{A}(\kappa)\right\|\cdot\|\mathbf{x}(k_0)\| \leq M_A{}^{k}\|\mathbf{x}(k_0)\|, \tag{129}$$

where $M_A=\sup\{\|\mathbf{A}(k\text{-}1)\|\}$, then since due to the normalization of terms in $\mathbf{A}(k\text{-}1)$ if $M_A<1$, as $\mathbf{x}(k)\rightarrow\mathbf{x}(k_0)$ *for* $k\rightarrow\infty$ about the point $\mathbf{x}(k_0)$ the system (108) is locally asymptotically stable. With respect to Definition 2, the term $\left\|\prod_{\kappa=k_0}^{k-1}\mathbf{A}(\kappa)\right\|\cdot\|\mathbf{x}(k_0)\|$ qualifies as a κL class function. Moreover, via setting $\mathbf{x}(k_0)=0$ we may analysis the conditions of the particular solution yielding a class κ_∞ function, where the following inequality should be satisfied further for BIBS

$$\|\mathbf{x}(k)\| \leq \left\|\sum_{\kappa=k_0}^{k-1}\prod_{i=\kappa}^{k-1}\mathbf{A}(i)\cdot\mathbf{B}(\kappa)\mathbf{u}(\kappa)\right\| \leq M_B\sum_{\kappa=k_0}^{k-1}M_A{}^{k-\kappa}\|\mathbf{u}(\kappa)\|, \tag{130}$$

where $M_B=\sup\{\|\mathbf{B}\|\}<\infty$. Therefore, if $\|\mathbf{u}(k-1)\|\leq L_u<\infty(k\geq k_0)$ then

$$\|\mathbf{x}(k)\| \leq M_B\sum_{\kappa=k_0}^{k-1}M_A{}^{k-\kappa}\|\mathbf{u}(\kappa)\| = \frac{M_B}{1-M_A}, \tag{131}$$

where the ratio $M_B/(1-M_A)=L_x$ for any $\|\mathbf{u}(k-1)\|\leq L_u$, $\mathbf{x}(k)\leq L_x<\infty$. Therefore according to [44], [85] the nonlinear discrete time state space system is BIBS stable. We may therefore recall that the above stability proof further applies to the proposed decomposed state space system (108), thus the following theorems can be stated

Theorem 3 (DDHS): The time-variant LMD of a HONU model or its extension as a nonlinear control loop i.e. $r;\gamma \geq 2$, in accordance to the representations (106) and (117)-(119) is locally BIBO stable, further locally asymptotically stable at the given state at sample k if

$$\rho(\hat{\mathbf{A}}(k-1)) < 1; \quad \rho(\hat{\mathbf{M}}(k-1)) < 1, \tag{132}$$

Theorem 4 (DDHS): The decomposed state-space representation (108) is BIBS stable for a single recurrent HONU, from initial sample time k_0 until k provided that

$$S = \left\|\hat{\mathbf{x}}(k)\right\| - \left\|\prod_{\kappa=k_0}^{k-1} \hat{\mathbf{A}}(\kappa)\right\| \cdot \left\|\hat{\mathbf{x}}(k_0)\right\| - \sum_{\kappa=k_0}^{k-1} \left\|\prod_{i=\kappa}^{k-1} \hat{\mathbf{A}}(i) \cdot \hat{\mathbf{B}}_{\mathbf{a}}(\kappa)\right\| \cdot \left\|\hat{\mathbf{u}}_{\mathbf{a}}(\kappa)\right\| \leq 0, \tag{133}$$

where multiplications of local matrix of dynamics $\hat{\mathbf{A}}(.)$ leads to the corresponding state transition matrices and k_0 is the initial stability evaluation sample time index. Then considering the control loop decomposed state space representation in (117)-(119), the BIBS stability of an MRAC-HONU control loop, from initial sample time k_0 until k, is justified if

$$S = \left\|\hat{\mathbf{x}}(k)\right\| - \left\|\prod_{\kappa=k_0}^{k-1} \hat{\mathbf{M}}(\kappa)\right\| \cdot \left\|\hat{\mathbf{x}}(k_0)\right\| - \sum_{\kappa=k_0}^{k-1} \left\|\prod_{i=\kappa}^{k-1} \hat{\mathbf{M}}(i) \cdot \hat{\mathbf{N}}_{\mathbf{a}}(\kappa)\right\| \cdot \left\|\hat{\mathbf{u}}_{\mathbf{a}}(\kappa)\right\| \leq 0, \tag{134}$$

where the state transition matrix correspondingly utilizes the local matrix of dynamics $\hat{\mathbf{M}}(.)$ ((122)-(123)).

Remark 4: From the theorems 3 & 4, in order that the DDHS method is valid in accordance to the theorem of ISS based stability in (125) the following restrictions yield:

i) The function defined as $\beta(\left\|\bar{\mathbf{x}}(k_0)\right\|) = \left\|\prod_{\kappa=k_0}^{k-1} \hat{\mathbf{A}}(\kappa)\right\| \cdot \left\|\hat{\mathbf{x}}(k_0)\right\|$ for $\forall k_0 < k < \infty$ must be asymptotically stable to remain as a κL class function.

ii) The asymptotic stability of $\beta(\left\|\bar{\mathbf{x}}(k_0)\right\|)$ in i) can be justified if $\rho\left(\left\|\prod_{\kappa=k_0}^{k-1} \hat{\mathbf{A}}(\kappa)\right\|\right) < 1, \ \forall k_0 < k < \infty.$

Theorem 5 (Strict DDHS): Provided that the conditions (133) & (134) for BIBS stability hold then from an initial time k_0 the BIBS stability is more strictly guaranteed if $S(k) \leq S(k-1) \ \forall 0 < k < \infty$ i.e.

$$\Delta S(k)=S(k)-S(k-1)\leq 0 \;\; \forall 0<k<\infty, \tag{135}$$

where considering setting of $k>k_0=k-3$ in the expression (133), the strict DDHS condition may be obtain as

$$\begin{aligned} S(k)-S(k-1)=&\|\hat{\mathbf{x}}(k)\|-\|\hat{\mathbf{x}}(k-1)\|-\left\|\prod_{\kappa=k-3}^{k-1}\hat{\mathbf{A}}(\kappa)\right\|\cdot\|\hat{\mathbf{x}}(k_0)\|+\left\|\prod_{\kappa=k-3}^{k-2}\hat{\mathbf{A}}(\kappa)\right\|\cdot\|\hat{\mathbf{x}}(k_0)\| \\ &-\sum_{\kappa=k-3}^{k-1}\left\|\prod_{i=\kappa}^{k-1}\hat{\mathbf{A}}(i)\cdot\hat{\mathbf{B}}_{\mathbf{a}}(\kappa)\right\|\cdot\|\hat{\mathbf{u}}_{\mathbf{a}}(\kappa)\|+\sum_{\kappa=k-3}^{k-2}\left\|\prod_{i=\kappa}^{k-2}\hat{\mathbf{A}}(i)\cdot\hat{\mathbf{B}}_{\mathbf{a}}(\kappa)\right\|\cdot\|\hat{\mathbf{u}}_{\mathbf{a}}(\kappa)\|. \end{aligned} \tag{136}$$

The condition is analogical for the closed loop system represented in (122)-(123). As can be seen the DDHS condition incorporates neighbouring process dynamics for given bounded process inputs, without being fixed to a given learning algorithm which is advantageous for practical application.

8 Experimental Analysis: Stability for HONU Adaptive Models and their Extension as a HONU-MRAC Control Loop

Following the theoretical background of HONUs and further the newly derived DHS and DDHS method, in this section a practical view into the use and properties of various HONUs (namely QNU and CNU, $r,\gamma=i,$ where $i=2,3.$ respectively) for adaptive identification and control is provided. As an extension of the work [90], this section demonstrates the validity and effectiveness of the DHS method based from the presented theories of BIBO, BIBS and Lyapunov stability criteria and further compares its results within defined limits of the DHS method to validate the DDHS method. Several practical applications are implemented on real-time engineering processes in the proceeding chapter, to justify its use for real industrial applications.

8.1 Nonlinear Position Feedback System Stability Analysis

To illustrate the performance of HONU adaptive identification and control to practical nonlinear system applications, further validation of the DHS and DDHS methods for application to HONU architectures. Let us consider a position feedback system as a modification of [84] described by the following relation

$$\begin{aligned}\dot{E} &= -x_2 \\ \dot{x}_2 &= \alpha\left[-\frac{1}{T}x_2 + \frac{f(E)}{T}\right].\end{aligned} \tag{137}$$

An intriguing property of this system lies within the dependence of the conditional gain α in the state variable equation $\dot{x}_2$. On selecting a suitable Lyapunov function describing a resulting energy function as follows

$$V = \frac{T}{2}x_2^2 + \int_0^E f(\partial)d\partial, \tag{138}$$

where the time constant $T=1$ and u described as a $f(E)$ as the system input, with an incorporated input saturation in range ± 0.5. Here, if $E = x_2 = 0$ the condition $V = 0$ is satisfied from the function(138). From this it follows that

$$\begin{aligned}\dot{V} &= Tx_2\dot{x}_2 + f(E).\dot{E} \\ &= Tx_2\left[-\frac{\alpha}{T}x_2 + \frac{\alpha.u}{T}\right] + u\cdot(-x_2) \\ &= -\alpha.x_2^2 + u\cdot(x_2)\left[\alpha - 1\right] \ \ where \quad \dot{V} \le 0 \quad if \\ 0 &\ge x_2 \ge \frac{u.[\alpha-1]}{\alpha}.\end{aligned} \tag{139}$$

Thus, the resulting condition of the derivative of the Lyapunov function yields that if $0 \ge x_2 \ge u.[\alpha - 1]/\alpha$ that the system is within a region of stability about the zero-equilibrium point. In this example a QNU is chosen featuring four previous output values of the identified neural model $n_y = 4$ and three previous input values of the process $n_u = 3$, trained via the GD learning algorithm where $\alpha = 0.2$ for identification under stable system dynamics. As an initial, two separate models are derived. As seen in Figure 34, a well identified QNU trained via the GD algorithm (t < 140[s], $\mu = 0.004$) is compared with perturbed neural weights (via multiplication of the well derived neural weight by the factor 1.22), introduced from t >140[s] on the process data, incorporating the same model configuration. From Figure 34, it is evident that the HONU model with perturbed neural weights encompasses unstable behaviour upon reapplication on newly fed input data from t>140 [s].

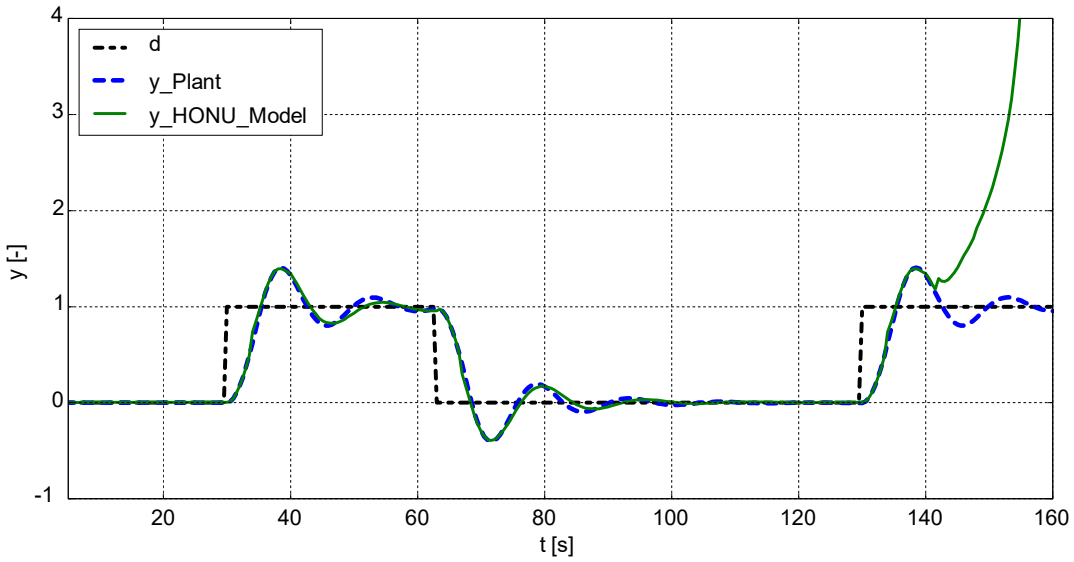


Figure 34 – HONU model performance of a well identified (stable, $\mu = 0.004$) QNU plant model in region t < 140[s] and perturbed neural weights (unstable, $1.22 \cdot \mathbf{w}$) in region of t >140[s] of the nonlinear position feedback system. For stability analysis refer to Figure 35-Figure 37.

As hypothesised Figure 35 a) illustrates the corresponding eigenvalues of $\bar{\mathbf{A}}$ for the unstable HONU model, compared with b) illustrating the eigenvalues $\bar{\mathbf{A}}$ corresponding to the well identified HONU model. From this diagram it is evident that the stable HONU model features all eigenvalues within the unit circle, contrary to the unstable HONU model where a clear positioning of the eigenvalues beyond the region of stability is evident. Given this result, the first remark of the DHS condition (88) yields that for the HONU model applied in time t<140[s] the BIBO stability of the model is satisfied for states in vicinity of the upper steady-state value.

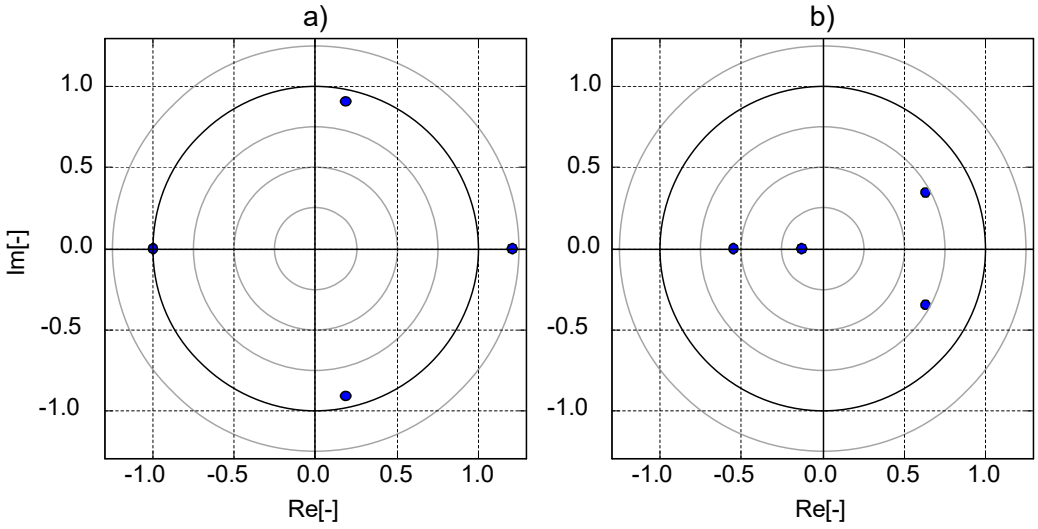


Figure 35 – DHS: Dispersion of respective HONU model eigenvalues for a Position Feedback System (Figure 34) a) Unstable HONU Model t>140[s] b) Stable HONU model t<140[s], derived about the upper steady state region. Concept presented in [90] and [107].

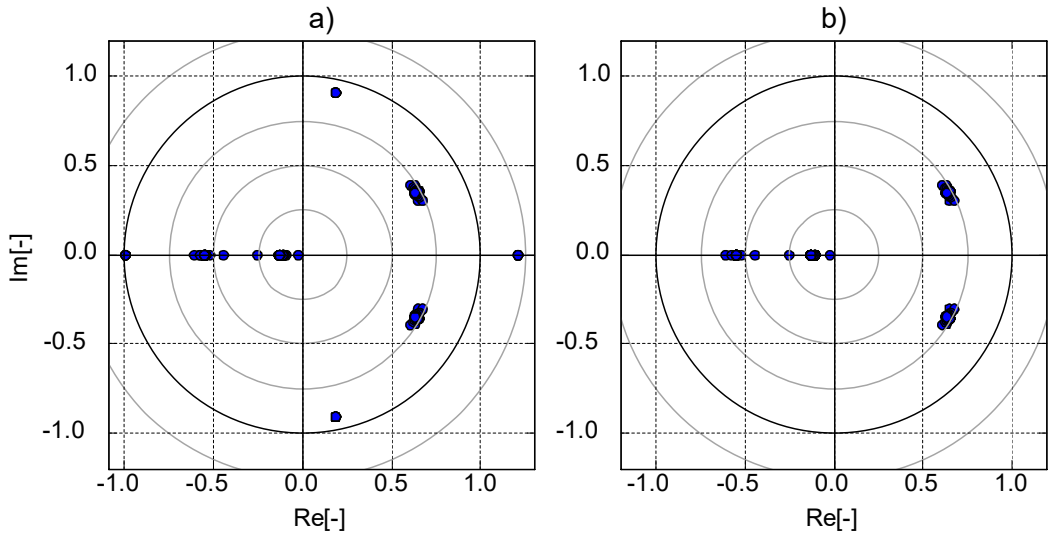


Figure 36 – Dispersion through time of respective HONU model LMD eigenvalues for a Position Feedback System (Figure 34) a) Unstable HONU model t=130-150[s] b) Stable HONU model. Concept presented in [90] and [107].

In accordance with the DDHS method, an evaluation of the local matrix of dynamics (LMD) of the HONU model $\hat{\mathbf{A}}(k)$ eigenvalues about each respective state point throughout the set of process operational points in time region t<140 [s], all eigenvalues are within the unit circle, contrary to that of the unstable HONU model applied from t>140 [s]. Thus, in accordance with the remarks of the DDHS method, the respective HONU model in region t<140[s] is justified to be locally BIBO stable for states within region of the upper steady-state point. From analysis of the BIBS inequality condition (133).

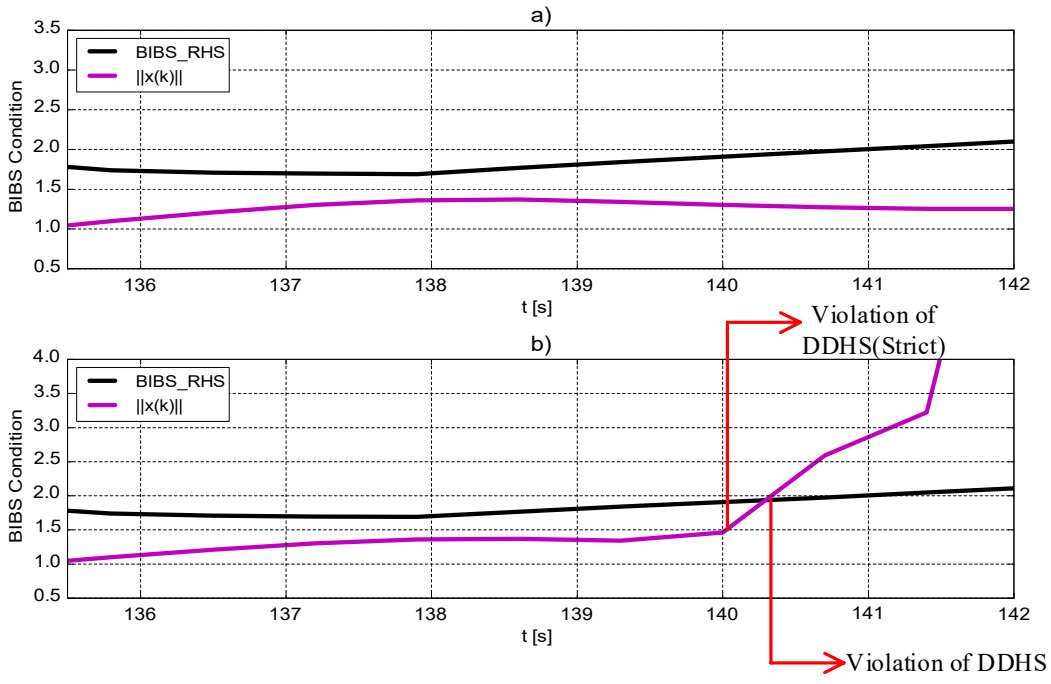


Figure 37- DDHS: Evaluation through time of BIBS condition (133), for Figure 34 a) stable and b) unstable HONU model from t>130[s] in Figure 34. The right-hand-side (BIBS-RHS) of (133) is less than the normalised state vector norm as t>140[s], hence violating the condition of BIBS stability. Concept presented in [90] and [107].

Figure 37 a), the HONU normalised state vector norms are evidently less than the corresponding state transition matrix conditions at time sample k across the applied region of t >130 [s], for the derived stable HONU model parameters. However, in Figure 37 b) the respective normalised state vector norms $\|\hat{\mathbf{x}}(k)\|$ at each time sample k yield values greater than the rate corresponding to the state transition matrices for t >140 [s], thus, a clear violation of the BIBS stability is seen. Figure 38 illustrates the performance of a tuned LNU model with extension of a LNU and QNU feedback controller separately (of the same parameters and RLS learning algorithm) where both the plant and controller feature $n_y = 4$ and previous HONU model output values and $n_u = 3$ previous process inputs. From Figure 38 b) it is intriguing that although a recurrent (dynamical) LNU is used as a single controller in feedback on a non-linear process. Due to the rather weak non-linearity of variable constant $\alpha = 0.2$, under incremental training via RLS through time, the LNU-LNU control loop model performs very well in comparison to a more appropriate non-linear (LNU-QNU) control loop variant.

Following analysis of the performance of a HONU model under both a stable and unstable set of HONU model neural weights, as a further study we may analyse the performance of the DHS and DDHS method on extension of a HONU-MRAC closed control loop. Figure 38 thus depicts the performance of a well identified HONU model with extension of a HONU feedback controller where $\mu = 0.9998$ (i.e. HONU-MRAC closed control loop), with exception of the time interval t=70-85 [s] where a second set of poorly identified HONU-MRAC controller weights are applied, identified with a rapid increase of learning rate $\mu = 0.9821$. From Figure 38 under stable control, the derived HONU-MRAC control loop exhibits excellent control performance in adhesion to the desired reference model of the position feedback system whilst featuring optimised speed in the process convergence to the setpoint value. In contrast, it is clear that the poorly identified HONU Model with extension of HONU feedback control exhibits an unstable response as applied from a zero-state region. Given this, our natural hypothesis is with regards to the position of the HONU-MRAC control loop eigenvalues corresponding to violation of local BIBO stability can be seen via analysis of the LMD eigenvalues of $\bar{\mathbf{M}}$ in Figure 38 c) and d) respectively.

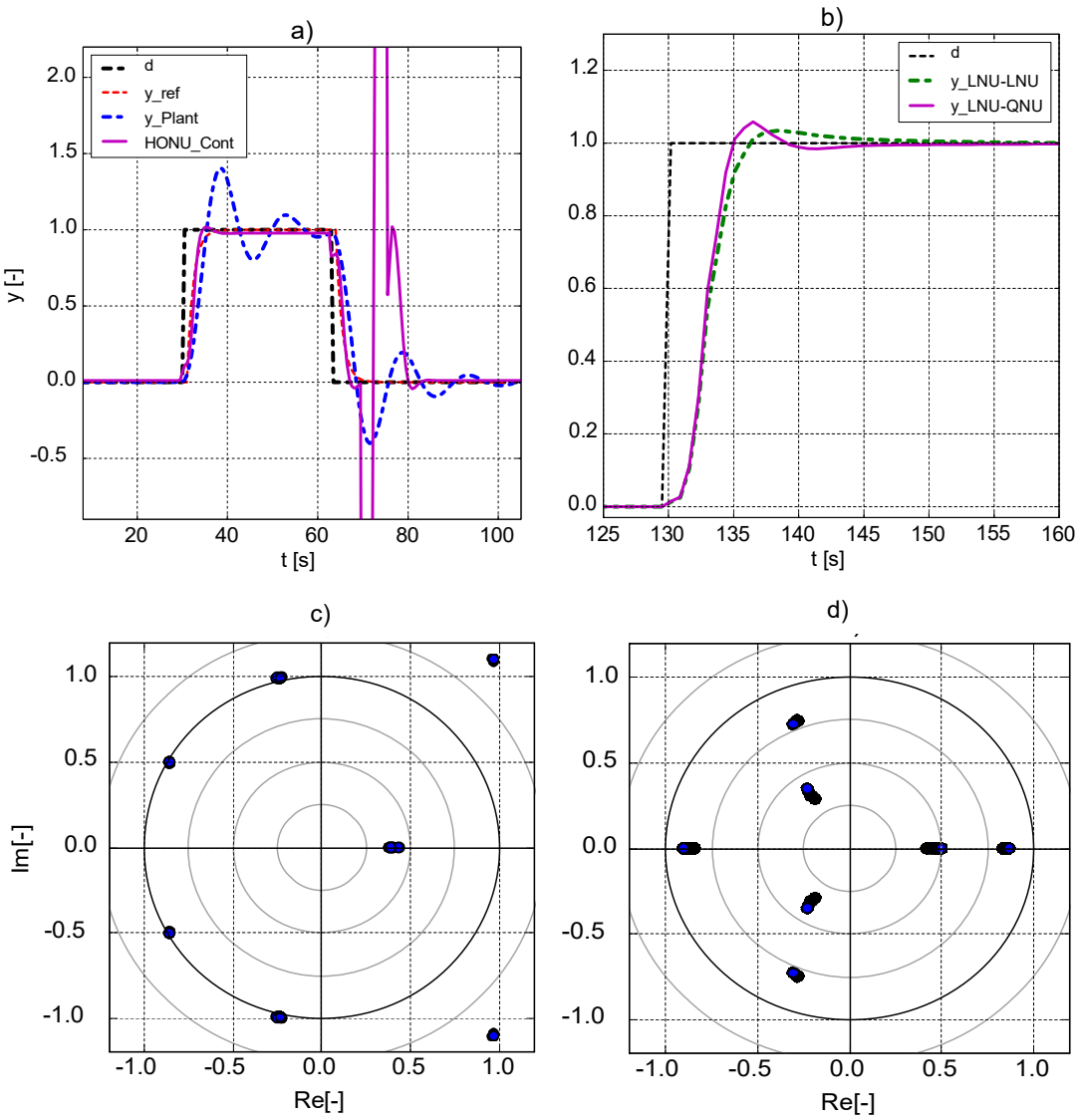


Figure 38 - a) LNU-QNU control loop trained both via RLS and introduced instability t=70-85 [s]. b) LNU and QNU controller comparison c) LMD eigenvalues at t=0-85 [s] (with poorly identified neural weights) d) t=85-200 LMD eigenvalues (with well identified neural weights). Presented in [89].

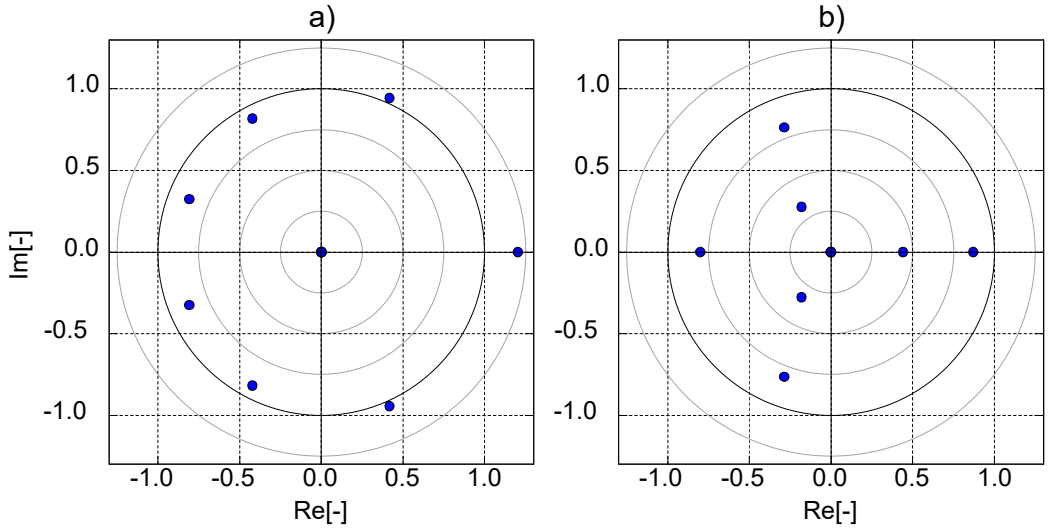


Figure 39- DHS: Dispersion of respective HONU model eigenvalues for a Position Feedback System (Figure 38) a) Unstable HONU model with HONU feedback controller (t=70-85 [s]) b) Stable HONU model with HONU feedback controller, evaluated about the initial state. Concept presented in [90] and [107].

For validation of local BIBO stability Figure 39 a) and b) illustrate the corresponding eigenvalues of the matrix of dynamics $\bar{\mathbf{M}}$ about the initial zero value region. Here, it is evident that on analysis of the time-invariant matrix of dynamics from t=70-85 [s] a violation of the BIBO stability is seen, which corresponds similarly to the results of the LMD $\hat{\mathbf{M}}$. Figure 40 a)-b) illustrates a sufficient simplification of the BIBS condition from (134) for repeated application of the bounded process or further, control inputs the given operating region as exampled in Figure 34. If the norm of the state transition matrix calculated across the defined operating region is converging to the threshold defined by the value B_1 i.e. $\left\| \prod_{\kappa=k_0}^{k-1} \hat{\mathbf{M}}(\kappa) \right\| \leq B_1$. Further, that the sum of norms of the STM as a product with the input matrix i.e. $\sum_{\kappa=k_0}^{k-1} \left\| \prod_{i=\kappa}^{k-1} \hat{\mathbf{M}}(i) \cdot \hat{\mathbf{B}}_{\mathbf{a}}(\kappa) \right\| \leq B_2$ holds, BIBS stability is sufficiently reasoned as the HONU-MRAC control loop response must be bounded (the values summarized in Table 7).

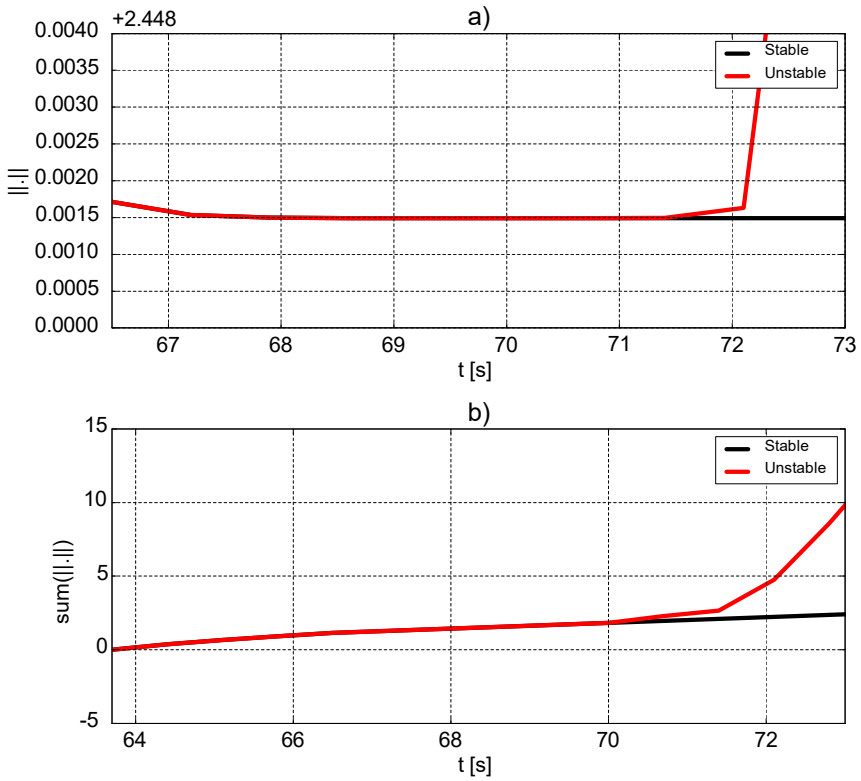


Figure 40- Evaluation through time of BIBS state transition matrix conditions from (134) to unstable HONU model with HONU feedback controller (t=70-85 [s]) (Figure 38). The respective matrix norms breach convergence of the derived boundaries for the given process region in Table 8. Concept presented in [90] and [107].

TABLE 7 STATE TRANSITION MATRIX (STM) NORM ANALYSIS OF DDHS		
Time Region t=65-75[s]	$\left\|\prod_{\kappa=k_0}^{k-1}\hat{\mathbf{M}}(\kappa)\right\|$	$\sum_{\kappa=k_0}^{k-1}\left\|\prod_{i=\kappa}^{k-1}\hat{\mathbf{M}}(i)\cdot\hat{\mathbf{B}}_{\mathbf{a}}(\kappa)\right\|$
Stable HONU	2.44948	3.34162
Unstable HONU	3.44954	12.8635

In Figure 41, confirmation of these results can be seen where the BIBS condition (134) is evaluated separately as a rate of state transition matrices (RHS) in comparison with the normalized state variable vector $\|\hat{\mathbf{x}}(k)\|$.

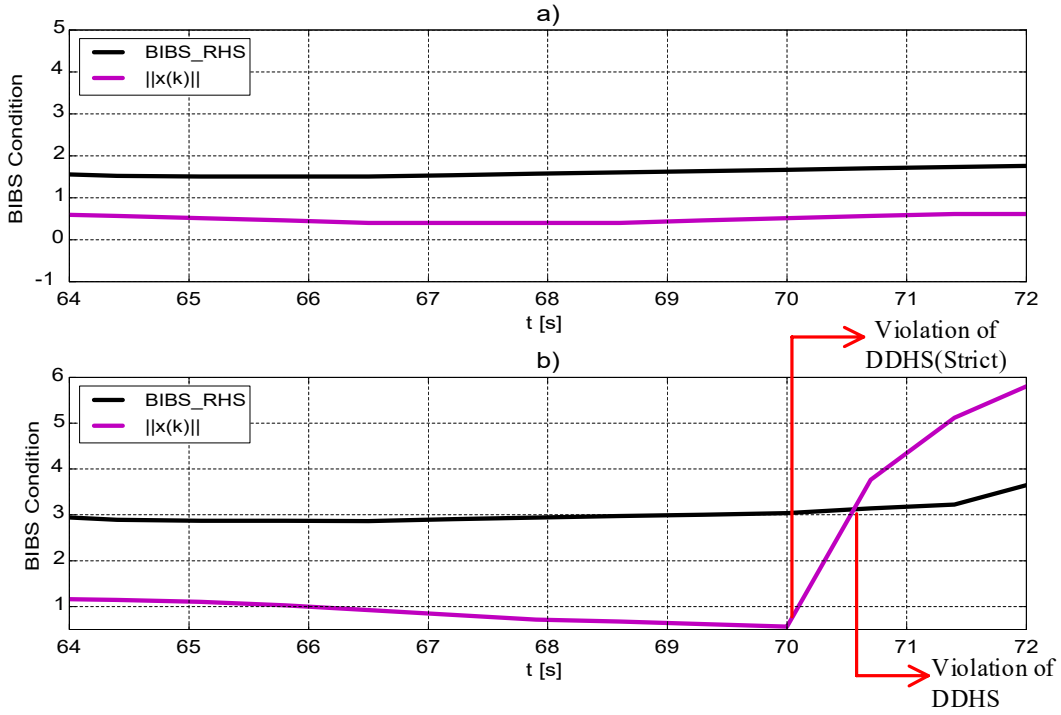


Figure 41-DDHS: Evaluation through time of BIBS condition (134), for stable and unstable HONU-MRAC control loop (Figure 38). Due to unstable HONU model with HONU feedback controller (t=70-85 [s]). S function evaluated without state variable vector norm of (134) is less than the normalised state vector norm as t>70[s], hence violating BIBS stability. Concept presented in [90] and [107].

However due to the relation (134) accounting for the previous samples of HONU-MRAC state transitions, it justifies the trajectory in state space for the given control input is becoming unstable as opposed to the local dynamics in vicinity of the discrete state point. In such case, though the LMD eigenvalues may locally recover i.e. $\rho(.)\leq 1$ in the sense of an adaptive control loop, however in a global sense the HONU-MRAC response may be dynamically unstable with respect to transition from neighbouring states. Thus, from the above analysis we may conclude that the DDHS method is indeed a functional means for verification of BIBS stability of a HONU model and its application as a non-linear control

loop. This was verified by analysis of the DHS in the same operating regions and as a further this dissertation analyses investigates its validity on several non-linear liquid tank systems as presented in the proceeding section.

8.2 Nonlinear Two-Funnel Tank System Stability Analysis

As a further example a nonlinear two-funnel tank system is considered. Studied from the work [86], the system may be described via the following balancing equations of motion

$$\begin{aligned}\frac{dh_1}{dt} &= \frac{4H^2}{\pi D^2 {h_1}^2}\left[q_{1f} - k_1\sqrt{|h_1 - h_2|}\right] \\ \frac{dh_2}{dt} &= \frac{4H^2}{\pi D^2 {h_2}^2}\left[q_{2f} - k_1\sqrt{|h_1 - h_2|} - k_2\sqrt{h_2}\right],\end{aligned} \tag{140}$$

where $H = 2.5\text{m},\ D = 1.5\text{m}$ valve constant $k_1 = 0.316\text{m}^3/\text{min}$, and $k_2 = 0.296\text{m}^3/\text{min}$. With regards to the system inputs, the variable q_{2f} corresponds to the inlet flow rate of the second funnel tank where the inlet flow rate into the first tank q_{1f} is considered constant at 0.1999 m^3/min, with the system output being the second tank height h_2. As an initial the stability of the system about the zero-equilibrium state may be analysed on selection of a suitable Lyapunov function candidate

$$V(h_1,h_2) = \frac{1}{4}\left[\int_0^{h_1} (\sigma_1)^2 d\sigma_1 + \int_0^{h_2} f(\sigma_2)^2 d\sigma_2\right], \tag{141}$$

where we may satisfy that about the states $h_1 = h_2 = 0,\ V(h_1,h_2) = 0$. Further for $\dot{V}(h_1,h_2) < 0$, it yields that

$$\begin{aligned}\dot{V}(h_1,h_2) &= \frac{1}{4}\left[\frac{dh_1}{dt}{h_1}^2 + {h_2}^2\frac{dh_2}{dt}\right] \\ &= \frac{H^2}{\pi D^2}\left[q_{1f} - 2k_1\sqrt{|h_1 - h_2|} + q_{2f} - k_2\sqrt{h_2}\right] < 0,\end{aligned} \tag{142}$$

therefore,

$$q_{2f} + q_{1f} > -\frac{2k_1\sqrt{|h_1 - h_2|}}{k_2\sqrt{h_2}}. \tag{143}$$

where from (143), given valve constants k_1 and k_2 are always positive, it is evident that the condition $\dot{V}(h_1,h_2)<0$ is satisfied across any positive inputs. Given this analysis, the example is focussed rather on the HONU-MRAC closed loop stability analysis with further insight to the influence on adaptive gain parameter p on the overall control loop stability. As an initial, Figure 42 depicts adaptive identification on the standalone nonlinear two-funnel tank system. Here, a disproportional rate of convergence to steady state for the various process inputs can be seen with various degrees of static gain with respect to the ratio of process inputs to outputs, which is dictated via the square relation of tank height h_2.

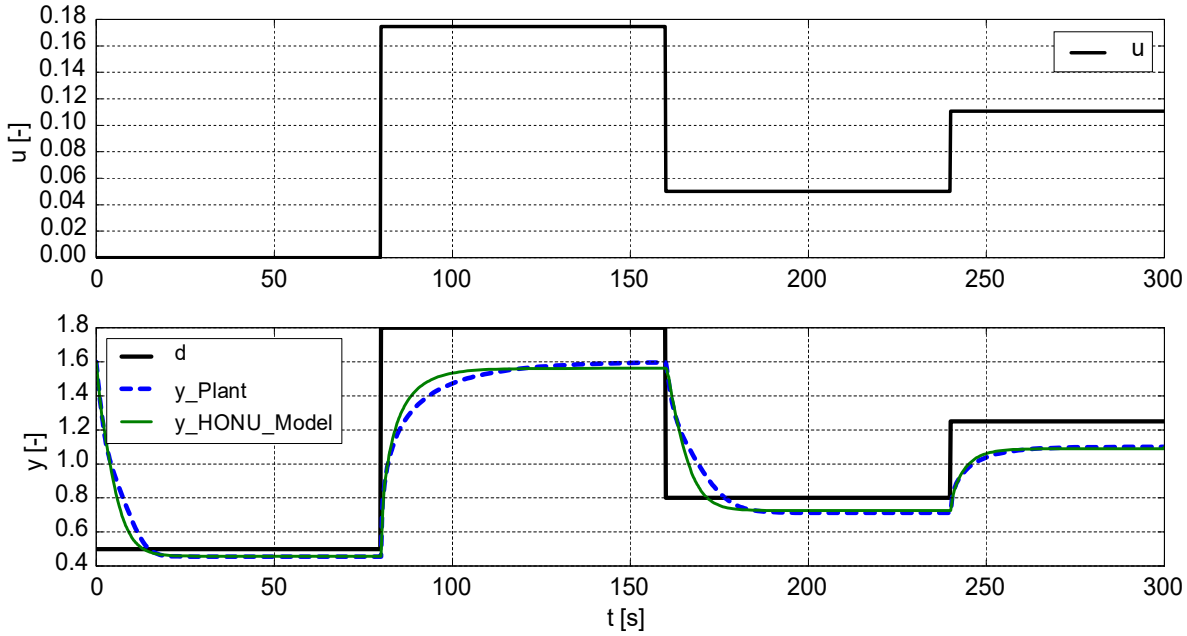


Figure 42 – Adaptive identification of standalone nonlinear two funnel tank system with QNU (HONU, r=2) trained via L-M algorithm $n_y = 6$, $n_u = 8$.

Two respective HONU-MRAC control loops are identified, the configuration parameters of which are summarized in table 8. To illustrate the performance of each derived HONU-MRAC control loop, a comparison is shown in Figure 43 with a conventional PID controller with the proportional gain setting of $r_o = 0.2 - 1.8$.

TABLE 8
HONU-MRAC CONTROL LOOP PARAMETERS

Type	**HONU**	**Setup**	**Pre-training**
LNU Plant + QNU Controller			
Plant	Recurrent LNU (r=1)	$n_y = 4$ $n_u = 5$	L-M (batch), 3000 epochs
Controller	Feedback QNU (r=2)	$n_y = 4$ $n_u = 5$	NGD (sample-by-sample), 100 epochs

QNU Plant + QNU Controller			
Plant	Recurrent QNU (r=2)	$n_y = 4 \quad n_u = 5$	L-M (batch), 1000 epochs
Controller	Feedback QNU (r=2)	$n_y = 4 \quad n_u = 5$	NGD (sample-by-sample), 100 epochs

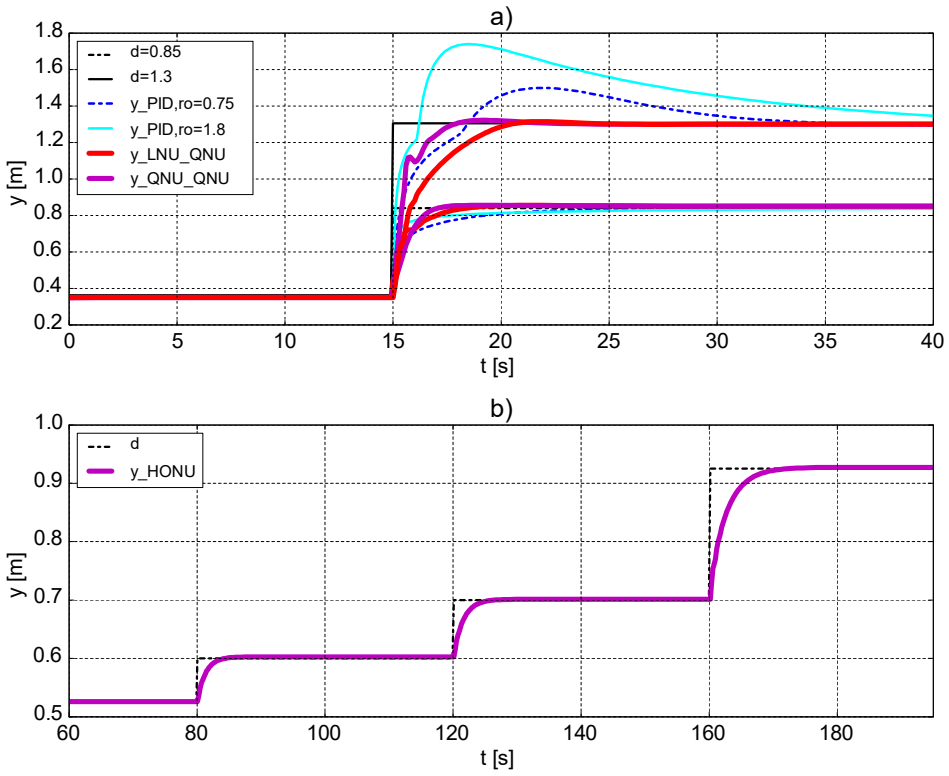


Figure 43 – Comparison of conventional PID controller with variable proportional gain and performance of the derived HONU-MRAC control loop. HONU-MRAC features fast dynamic response and better adhesion across the range of desired system operating points of the nonlinear two funnel tank system. Concept presented in [90] and [107].

Although the PID controller is able to adequately perform with minimal overshoot at lower liquid levels, a controller of the same parameter settings is unable to adequately achieve the desired speed of convergence and minimal steady state error for the whole range of identified process operating points. However, the derived QNU-LNU closed control loop derived is indeed able to achieve fast speed of convergence and minimised steady state error about all trained operating points, even as a standalone closed control loop of constant parameters, whilst also maintaining a similar computational efficiency due to its polynomial structure as compared with the conventional PID control algorithm. It is also intriguing that an LNU plant model trained via batch learning algorithms further applied with incremental training as a real-time control loop performs sufficiently well, in comparison to the faster QNU-QNU control loop configuration.

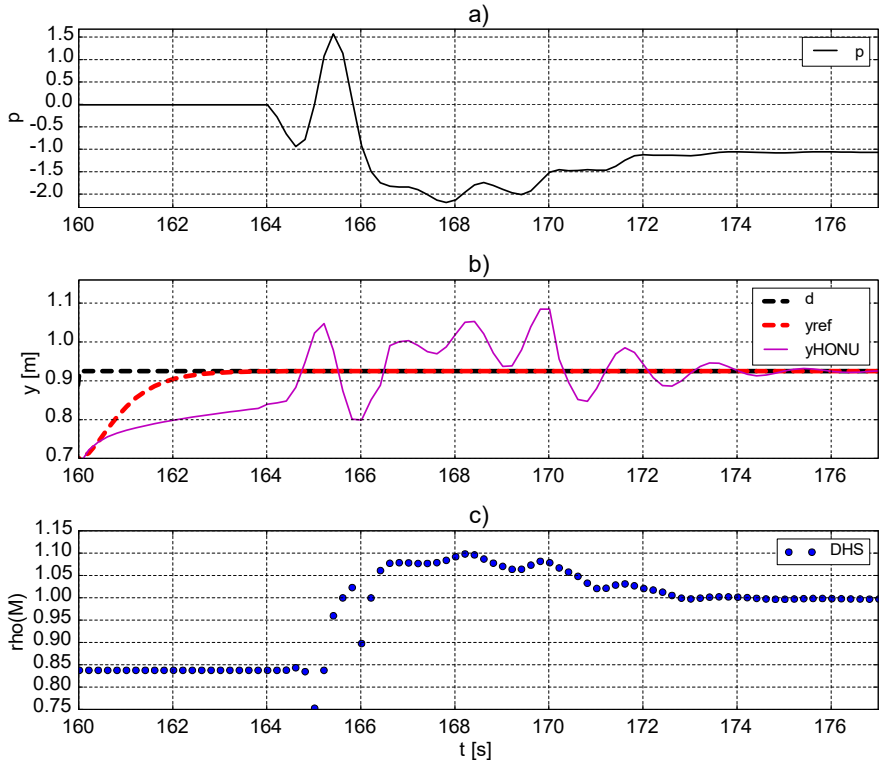


Figure 44- DHS: a) Randomly changed controller gain $p(t)$ in adaptive QNU-QNU control loop (instability onset $p(t=164.6)=-1.18$). b) Control loop output and the reference (desired) output. c) $\rho(t)$ computed via DHS. Concept presented in [90] and [107].

Following the design of the respective HONU-MRAC control loop, for the next stages of analysis let us consider the QNU-QNU closed control loop summarized in Table 6. To analyse validation of BIBS stability via the DDHS method, and further influence of the adaptive gain parameter p the HONU-MRAC control loop is initially applied with NGD training on the previously identified plant model dynamics from Table 6, using the pre-trained HONU feedback controller weights as an applied constant parameter control from t=0-150. The adaptive gain parameter $p=-0.01$ is constantly applied until t=150[s]. In the same data set (epoch) from t>150[s] the HONU-MRAC is applied as an adaptive parameter control loop where the gain parameter p is manually updated sample-by-sample in accordance to Figure 44 from t=164[s]. A corresponding analysis of the spectral radii $\rho(\bar{\mathbf{M}}(k))$, corresponding to the local BIBO stability via DHS method is further illustrated in Figure 44 b).

Figure 45 analyses the performance of the DDHS method via BIBS stability analysis. The HONU-HONU control loop (the same as in Figure 44) corresponding to $p(t<164)=-0.985$ and $p(t\geq 164)=-2.25$ are simulated. From this analysis, $p=-0.985$ provides the optimal performance, reducing the steady state error within several samples; however, an unstable

response develops after introducing $p=-2.25$ (Figure 45 a)), where the BIBS stability condition (134) shows the stability violation after $p(t=164.6)=-1.18$.

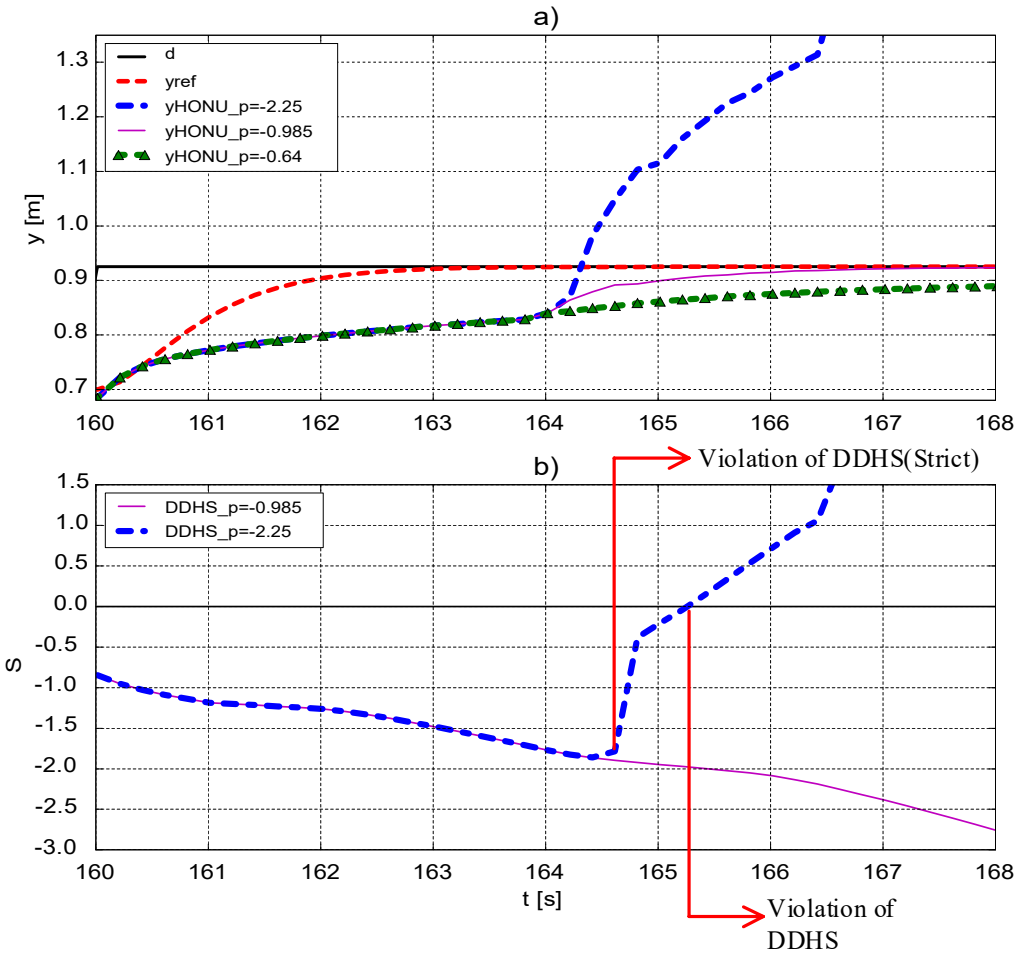


Figure 45 - DDHS: a) QNU-QNU control loop (as in Figure 44) responses with feedback controller gain $p(t>164)=-0.985$ *(stable) and* $p(t>164)=-2.25$ *(unstable). b) BIBS stability condition via DDHS is violated for* $p(t>164)=-2.25$. *Concept presented in [90] and [107].*

8.3 Two-Tank Liquid Level System Stability Analysis

As a continuation of the HONU-MRAC control loop study presented in section 5.1, let us recall the two-tank water level system, defined via the following mathematical relations

$$\begin{aligned} A\cdot\frac{dh_1}{dt} &= Q_t - C_{db}\cdot s_1\cdot\sqrt{2\cdot g\cdot(h_1-h_2)} \\ A\cdot\frac{dh_2}{dt} &= C_{db}\cdot s_1\cdot\sqrt{2\cdot g\cdot(h_1-h_2)} - C_{dc}\cdot s_2\cdot\sqrt{2\cdot g\cdot h_2}. \end{aligned} \tag{144}$$

Let us further consider the same design parameters where the tank cross-sectional area $A=0.002[m^2]$, the respective cross-sectional area of the orifices $s_1=s_2=0.000785[m^2]$, orifice discharge coefficients $C_{db}=C_{dc}=0.60$. Further, let us consider the fluid medium to be

water with density $\rho=1000[kg/m^3]$ and the gravitational constant of acceleration $g=9.81[ms^{-2}]$.

As the focus of this section is to analysis the stability of the presented HONU-MRAC control loop design presented in 5.1, let us consider the design of a HONU-MRAC control loop on a simulated version of the plant (144). In this case an LNU-QNU closed control loop is identified via RLS training with 5 previous model output values. 4 previous process inputs are further incorporated into the input vector. The HONU feedback controller consists of a single HONU feedback controller with the same input vector length i.e. $n_y=5$ and $n_u=4$, the feedback gain $p=0.01$ and sampling period 0.4 [s] (a comparison of the HONU-MRAC loop as an offline tuned and real-time constant parameter loop is shown in Figure 46).

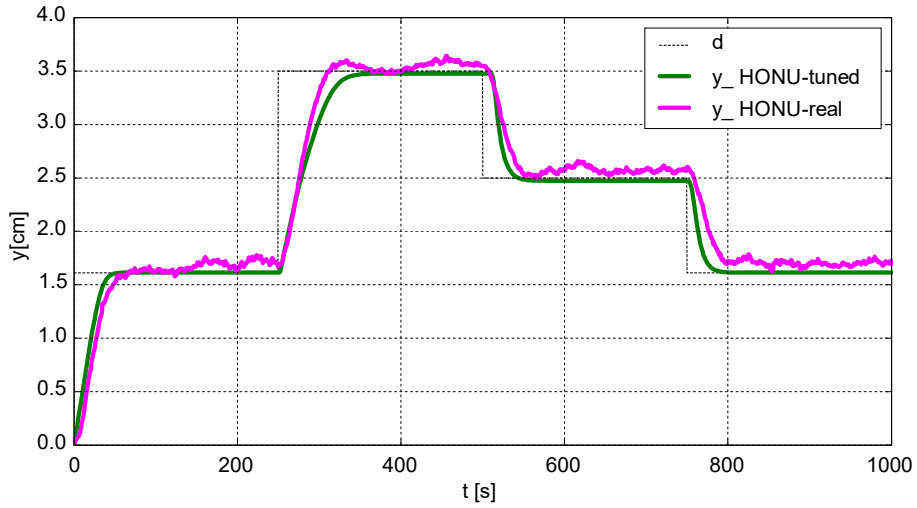


Figure 46-Real-time online HONU adaptive feedback control of two-tank water level system, compared with the offline tuned HONU-MRAC control loop response. As presented in [97], [107], [92] and further studied in [91].

To provide a preliminary assessment into stability, after 200 epochs of training, in the last training epoch a large increase in the learning rate is introduced to $\mu=0.9998$ at time t>488[s] on the simulated plant. From Figure 47 c) and d) it is evident that from t>490[s] the condition (134) switches from a monotonic decrease to $\Delta S>0$ and hence signifies the onset of instability where the BIBS condition is violated for t>492[s]. This is reflected in the Figure 47 b) via the violation of BIBO stability corresponding to spectral radii $\rho(.)>1$ however due to the relation (134) accounting for previous samples of HONU-MRAC state transitions, the strict DDHS condition (135) yields a stronger condition that clearly pronounces that onset of instability. It further justifies the trajectory in state space for the

given control input is becoming unstable as opposed to the local dynamics in vicinity of the discrete state point.

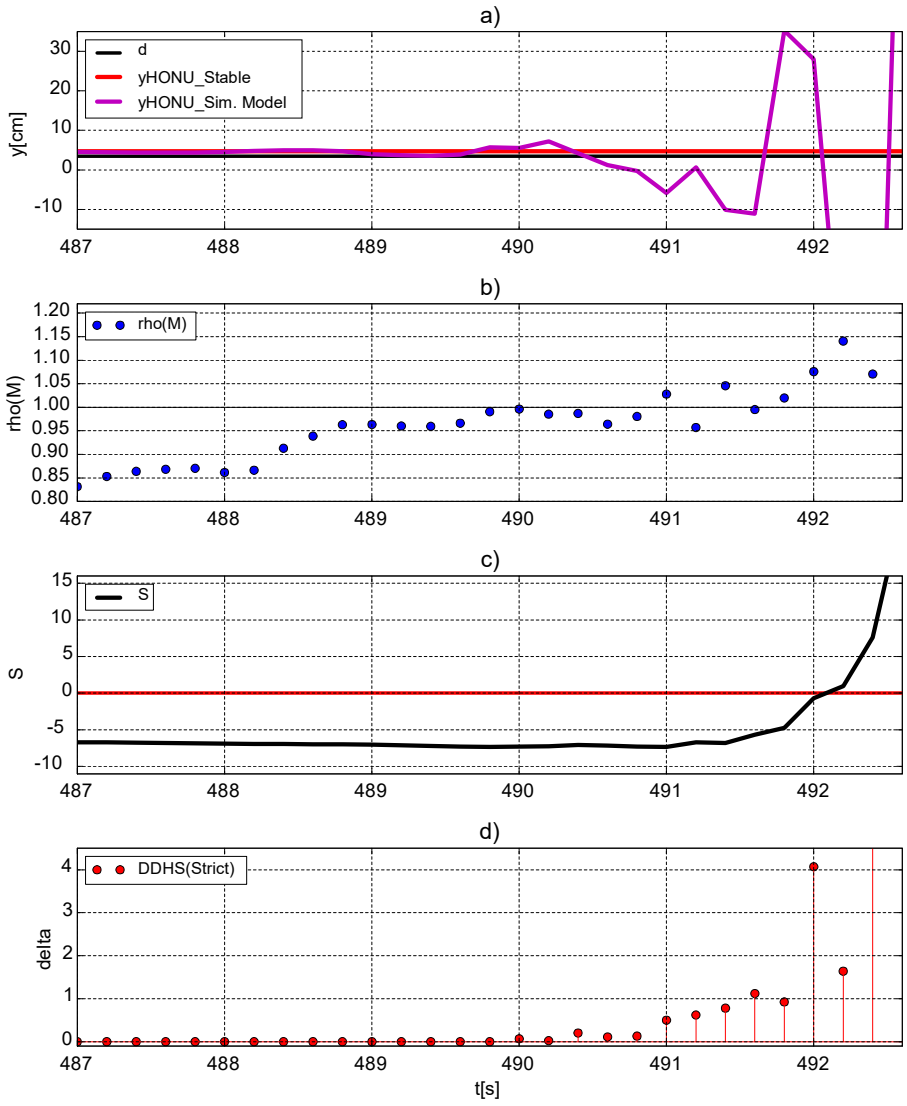


Figure 47 -a) Adaptive (RLS) LNU-QNU control loop becomes unstable soon after learning rate $\mu(t>488)=0.9998$ *.b) Spectral radii through time of LNU-QNU closed loop LMD c) BIBS condition (134) through time. d) Strict DDHS reveals onset of instability soon after learning rate becomes changed for* t>*488. Presented in [97] & [107].*

Following stability assessment on the simulated plant, following from Figure 46, we may also validate the whole LNU-QNU closed loop stability via the DHS and DDHS conditions. An intriguing result from the applied real-time implementation in Figure 46 is that for varying operating levels, such nonlinear HONU-MRAC control loop is capable to operate with similar performance where the natural static gains of the system differ, even as a constant parameter control loop. Following analysis of the simulated two-tank control loop stability, Figure 48 shows the computation of eigenvalues in region of each steady-state operating point (0, 1.5, 2.5 and 3.5 [cm]) for the real-time implementation of the HONU-MRAC control loop. Indeed, the BIBO stability for the linearized approximation of the LNU-QNU closed control loop is verified in accordance to the DHS condition.

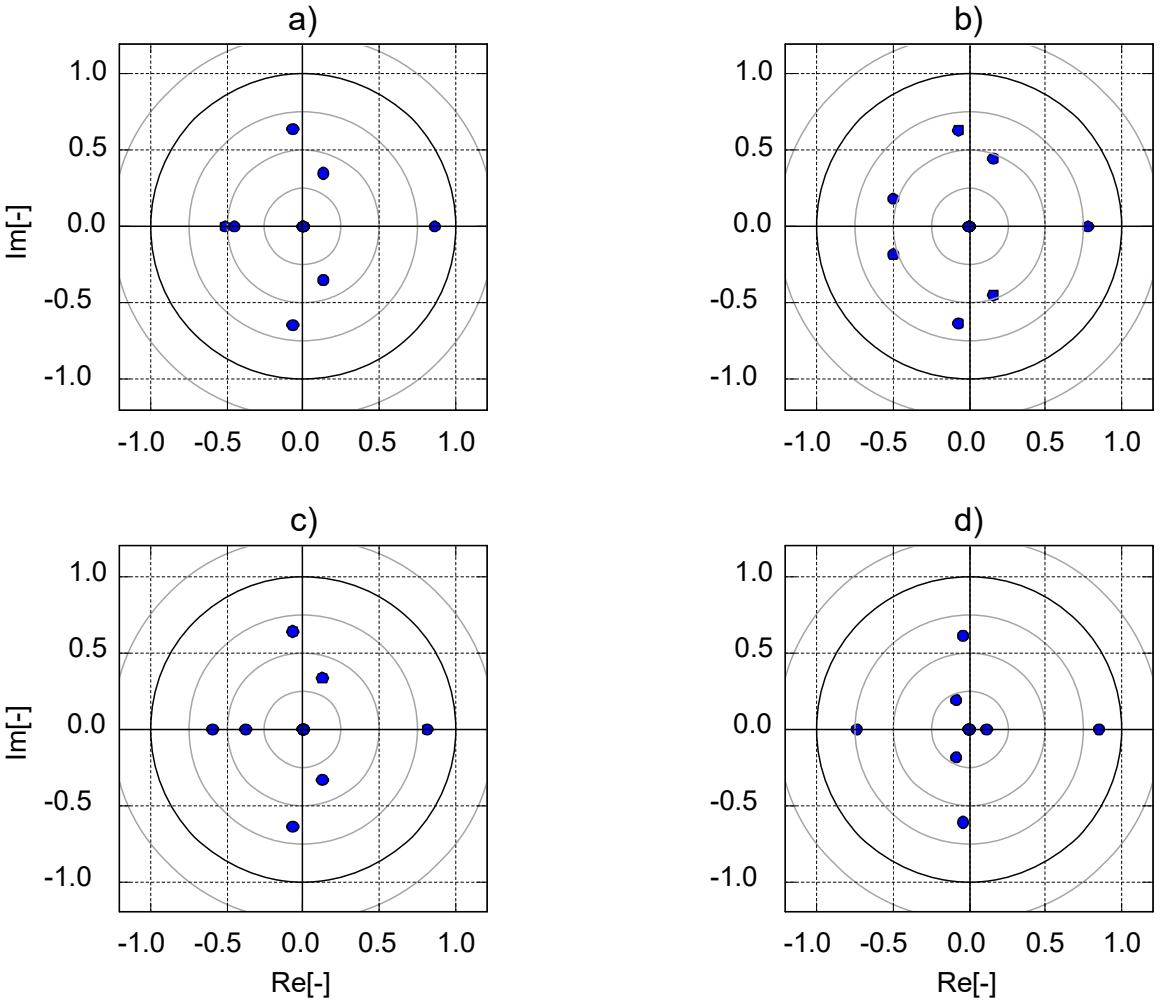


Figure 48- DHS: Dispersion of HONU model eigenvalues with extended HONU feedback control for real two-tank water level system (Figure 46), evaluated about a) 0 [cm], b) 1.5 [cm] c) 2.5 [cm] d) 3.5 [cm]. Concept presented in [90] and [107].

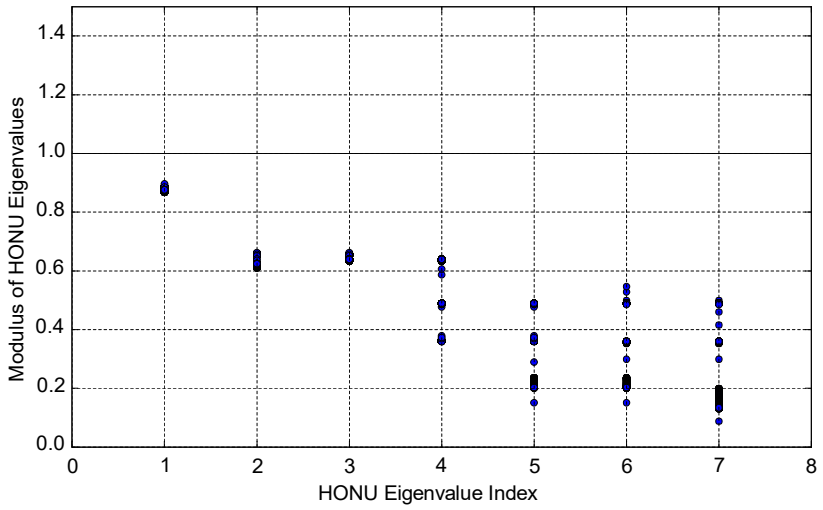


Figure 49 - Moduli of HONU model LMD eigenvalues with extended HONU feedback control for real two-tank water level system control (Figure 46). Concept presented in [90] and [107].

The results of Figure 49 further illustrate the local BIBO stability are satisfied via $\rho(.)<1$ across all operating points in each sample of $\hat{\mathbf{M}}(k)$. Figure 50 a) and b) analyse a comparison of the normalised state vector norms with the sum of state transition matrix norms from the BIBS condition (134) in time regions t=0-250 and t=255-500 respectively, where k_0 is selected at each change of the desired setpoint.

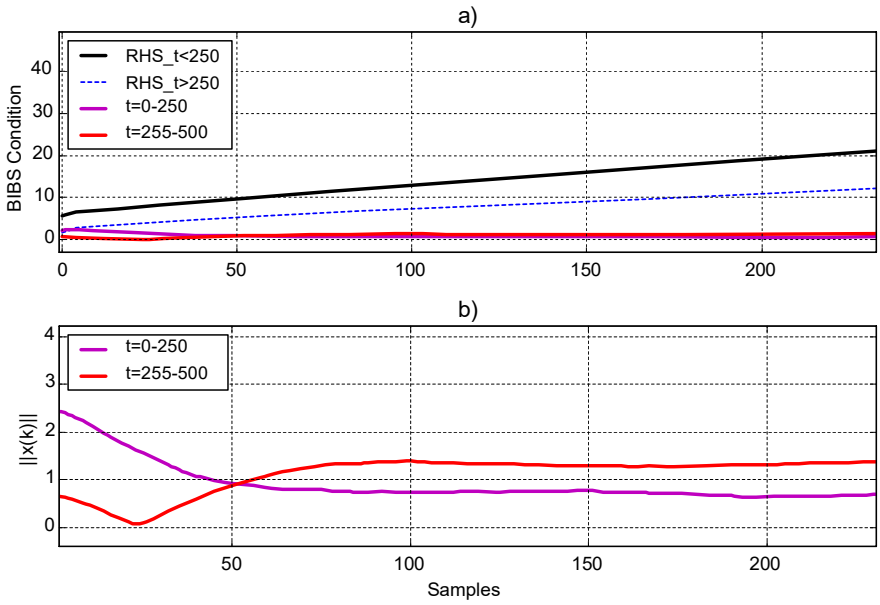


Figure 50- DDHS: Evaluation through time of BIBS condition(134) from Figure 46. a) Evaluation of BIBS condition (134) for t=0-250 and t=255-500 of HONU-MRAC closed control loop b) Detail of state vector norms. The sum of state transition matrix norms of (134) is greater than the state vector norm across both operating regions, respecting the condition of BIBS stability. Concept presented in [90] and [107].

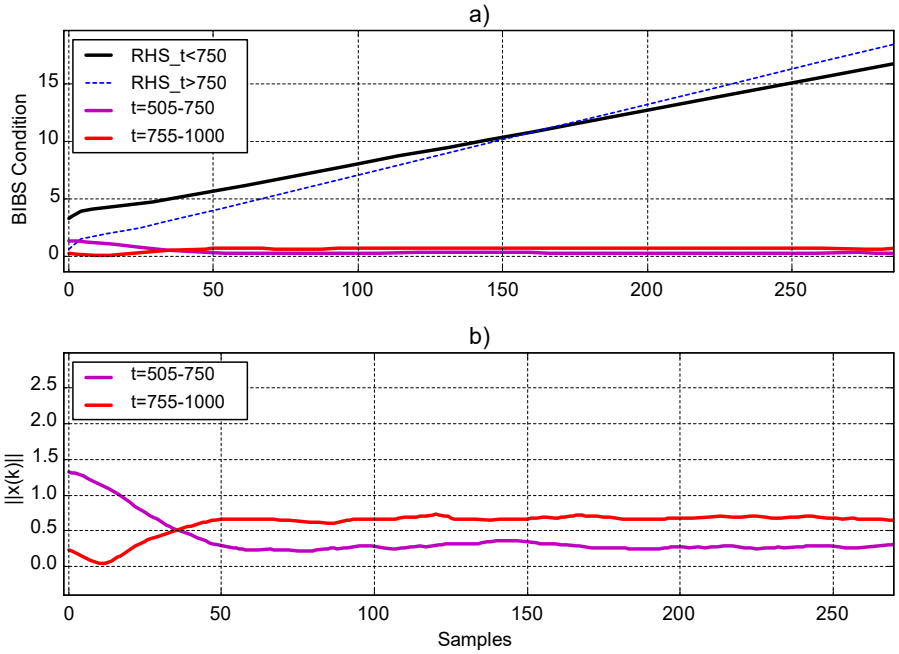


Figure 51- DDHS: Evaluation through time of BIBS condition (134) from Figure 46. a) Evaluation of BIBS condition (134) for t=505-750[s] and t=755-1000[s] of HONU-MRAC closed control loop b) Detail of state vector norms. The sum state transition matrix norms of (134) is greater than the state vector norm across both operating regions, respecting the condition of BIBS stability. Concept presented in [90] and [107].

From Figure 51, a similar conclusion yields with respect to evaluation over time regions t=505-750 and t=755-1000. From analysis across all state points of the normalised state vector norms in comparison to the sum state transition matrix norms of the BIBS condition (134), both respective time regions conclude BIBS stability is satisfied on application of the real-time constant parameter LNU-QNU control loop applied to the two-tank liquid level system.

8.4 Experimental Railway Stand (CTU Roller Rig) Stability Analysis

Following the study of HONU-MRAC adaptive control as applied to the CTU roller rig in section 5.2, this section provides a continuation via comprehensive stability analysis with the proposed DHS and DDHS methods for the main control results presented in section 5.2. As an additional means of validation, a Lyapunov inequality condition is derived and applied to compare the presented results following from the BIBS based condition derived in section 6.3.

8.4.1 HONU-MRAC Stability Analysis for Conventional Wheelsets

As this section aims to analyse the stability of the derived HONU-MRAC control loop as applied to a conventional wheelset configuration, let us consider the mathematical model of the roller rig as described in relations (30)-(35). Then, as a means of validation and comparison behind the newly proposed DDHS approach, let us derive a Lyapunov function based rule for assessing the adaptive control algorithm of the whole HONU-MRAC loop. Taking a modification of the Lyapunov function candidate presented in [20] we may define the following

$$V(k)=e(k)^2+\Delta\mathbf{v}(k)\Delta\mathbf{v}(k)^T, \tag{145}$$

given the relation (145), it is apparent that for Lyapunov function based stability the following difference should be evaluated

$$\begin{aligned}\Delta V(k)=V(k)-V(k-1)=e(k)^2+\Delta\mathbf{v}(k)\Delta\mathbf{v}(k)^T\\ -e(k-1)^2-\Delta\mathbf{v}(k-1)\Delta\mathbf{v}(k-1)^T.\end{aligned} \tag{146}$$

From (145) via RLS algorithm the following weight update rule may be defined

$$\mathbf{v}(k)=\mathbf{v}(k-1)+e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k). \tag{147}$$

Then it yields that,

$$\begin{aligned}\Delta V(k) = e(k)^2 + &\left[\mathbf{v}(k-1) + e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)\right]\\ &\cdot\left[\mathbf{v}(k-1) + e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)\right]^T - e(k-1)^2\\ &-\left[\mathbf{v}(k-2) + e(k-1)\cdot\mathbf{col}\xi(k-1)^T\cdot\mathbf{R}^{-1}(k-1)\right]\\ &\cdot\left[\mathbf{v}(k-2) + e(k-1)\cdot\mathbf{col}\xi(k-1)^T\cdot\mathbf{R}^{-1}(k-1)\right]^T.\end{aligned} \tag{148}$$

From the definition of Lyapunov stability, provided that $\Delta V(k) < 0$, and given the update rule (147) for $\mathbf{v}(k-2)$ it yields that for asymptotic stability

$$\begin{aligned}&e(k)^2 + \mathbf{v}(k-1)\cdot\left[e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)\right]^T +\\ &[e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)]\mathbf{v}(k-1)^T +\\ &[e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)]\cdot[e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)]^T\\ &-e(k-1)^2 < 0.\end{aligned} \tag{149}$$

Thus, the relation (149) may be reduced to the following inequality

$$\begin{aligned}&\mathbf{v}(k-1)\cdot\left[e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)\right]^T +\\ &[e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)]\mathbf{v}(k-1)^T +\\ &[e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)]\cdot[e(k)\cdot\mathbf{col}\xi(k)^T\cdot\mathbf{R}^{-1}(k)]^T\\ &< e(k-1)^2 - e(k)^2,\end{aligned} \tag{150}$$

then given each term is a scalar, taking the norm of both sides yields following generalized condition for asymptotic stability about a given state point

$$\begin{aligned}&\left\|\mathbf{v}(k-1)\cdot\Delta\mathbf{v}(k)^T + \Delta\mathbf{v}(k)\cdot\mathbf{v}(k-1)^T + \Delta\mathbf{v}(k)\cdot\Delta\mathbf{v}(k)^T\right\| <\\ &\left\|e(k-1)^2 - e(k)^2\right\|.\end{aligned} \tag{151}$$

Following derivation of the Lyapunov stability condition (145)-(151), the LNU-QNU control loop (as shown in Figure 18) is applied in real-time with RLS training. From Figure 52 a) at *t*=280 an increase in the learning rate is induced such to cause unstable set of derived neural weight in the feedback controller. Further during adaptation, the controller gain $p = 0.65$ is set constant. In Figure 52 a comparison with the derived Lyapunov stability condition (145)-(151) and strict DHS condition (135) validates the proposal of ΔS for monitoring of approaching instability of the whole control loop, before the control loop is highly unstable from *t*>319.

On comparison with (151) the condition $\Delta V < 0$ is again achieved in t=312[s] where the condition ΔS continually increases. This is also the advantage for practical application to monitor state transition based criteria which more globally encompasses the whole system dynamics with respect to neighbouring states as opposed to local state point stability as in the sense of the Lyapunov condition (151). Further, on monitoring real control loop stability via DDHS in Figure 54 b) it is clear that as long as the HONU-MRAC control loop is stable during adaptation, even with initially poorly tuned neural weights the BIBS condition monotonically decreases while control performance adaptively improves.

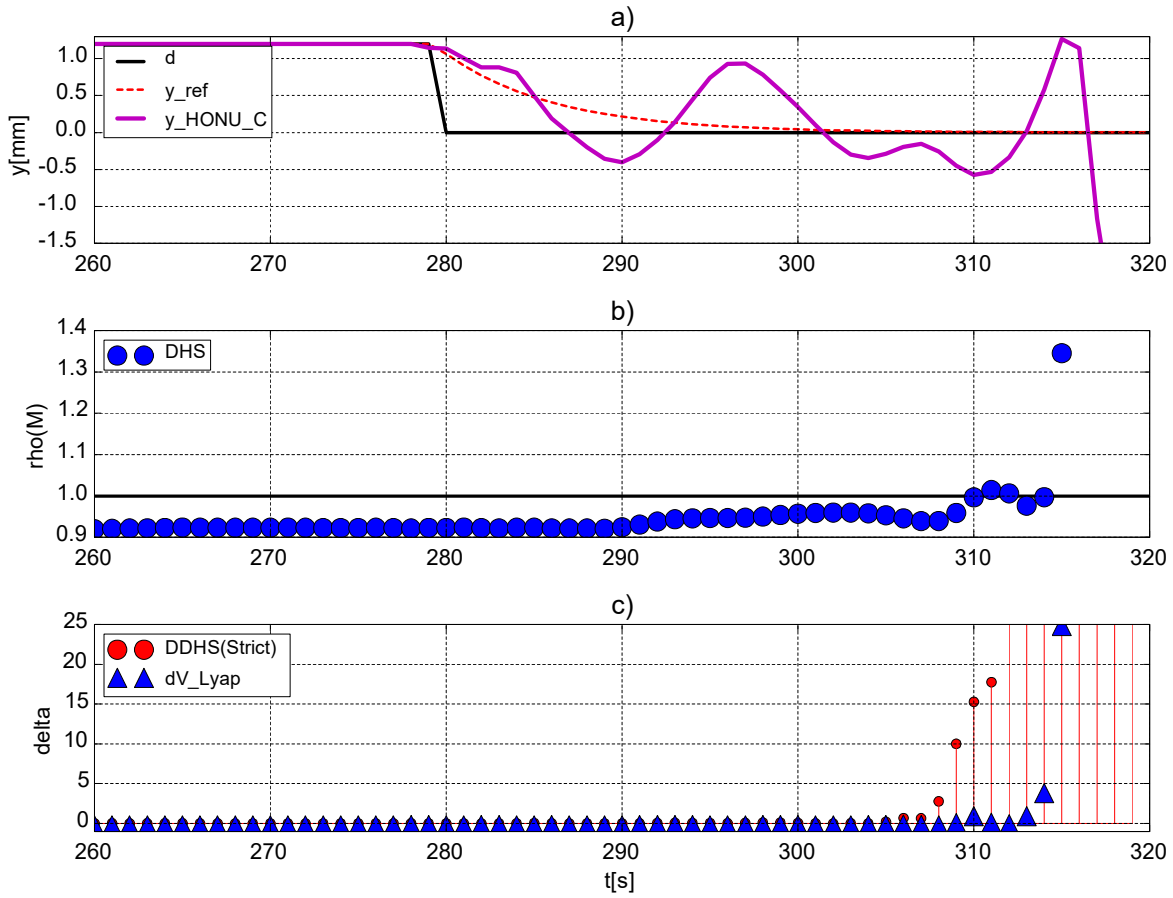


Figure 52: a) LNU-QNU control loop with RLS pre-training and adaptive GD in last epoch becomes unstable soon after learning rate $\mu(t > 280) = 0.01$ .b) Analysis of spectral radii through time (BIBO stability) via DHS c) Validation and comparison with Lyapunov condition (151) and strict DDHS (135) which reveals clearer condition for ongoing instability of whole LNU-QNU loop. Concept presented in [90], [107] and [106].

8.4.2 HONU-MRAC Stability Analysis on Real Roller Rig with IRW Configuration

With concern to any real-time adaptive control algorithm stability yields to be a key topic to ensure the derived control loop either, applied as a constant parameter configuration or, online adaptive form as in Figure 54 a) maintains dynamical stability along its trajectory in state space for the required operating points. As an initial, the time-invariant matrix of

dynamics $\bar{\mathbf{M}}$ eigenvalues are evaluated over the steady state regions of a) t=120-150[s], b) t=150-180[s] and c) 180-210[s].

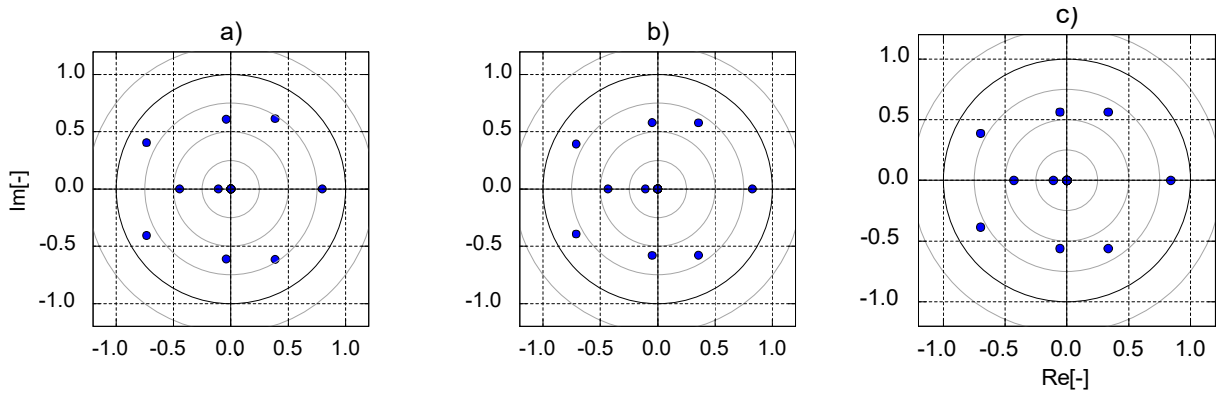


Figure 53 – DHS: Dispersion of GD tuned LNU-QNU closed control loop eigenvalues for real-time control of the roller rig system (Figure 24), evaluated about a) t=120-150 [s] (0 [mm]), b) t=150-180 [s] (2 [mm]) c) t=180-210 [s] (3 [mm]) respectfully. Concept presented in [90] and [107].

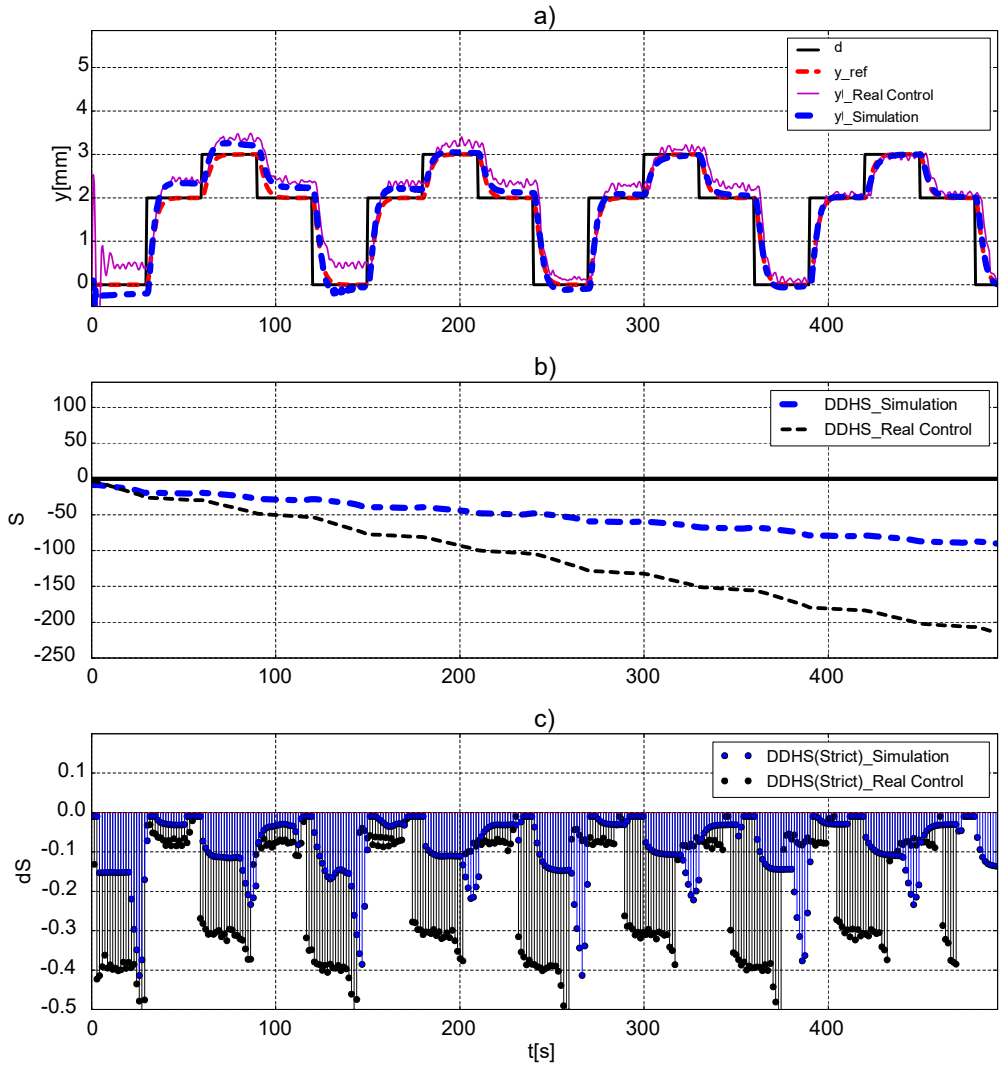


Figure 54 – a) Fully adaptive HONU-MRAC control loop via QNU plant trained via NGD and LNU feedback controller trained via RLS learning algorithm. $n_y = 4$, $n_u = 5$. QNU-LNU configuration optimises within 4 epochs, from changed bogie frame stiffness and damping dynamic properties. b) DDHS stability condition (for $k_0 = 0$). c) Strict DDHS confirms stability of the control loops Concepts presented in [88], [95] and [107].

In Figure 53 it is evident that all eigenvalues of the matrix of dynamics $\bar{\mathbf{M}}$ evaluated from the response Figure 26 respect local BIBO stability from (88). On analysing the HONU-MRAC closed control loop stability via the DDHS in Figure 54 b) –c), it is clear that as long as the HONU-MRAC control loop is stable during adaptation, even with initially poorly tuned neural weights, the BIBS condition monotonically decreases while control performance adaptively improves.

In this section the design and application of a HONU-MRAC closed control loop was investigated and successfully verified as a means of effective real-time control of the lateral skew of an experimental railway stand with IRW wheelset configuration for low to mid-range speed. Along with verification of the HONU-MRAC whole closed control loop stability both as a constant parameter closed control loop and via real-time controller parameter tuning. The evaluation via DHS justifies the results of local BIBO stability via DDHS evaluation. Furthermore, both the theoretical study of an unstable adaptive control loop applied to a convention wheelset configuration and further real-time adaptive control loop on a IRW configuration, verified the capabilities of the DDHS approach to detect an oncoming set of instability as well as validating real-time BIBS stability of the applied HONU-MRAC adaptive control loop. As a further to the results presented in this dissertation, an extension of this work for future research could be to analyse application under higher speed ranges as well as MISO or moreover, MIMO process configuration.

8.5 HONU-MRAC Design for Barrier Drive Control Stability Analysis

Following design of the HONU-MRAC closed control loop as presented in section 5.3, an analysis into the application of the DHS and further DDHS conditions may be used to justify the designed HONU-MRAC closed control loop stability during application about critical operating points of the barrier drive motion. For the scope of this section, the real-time results of Figure 32 corresponding to the HONU-MRAC closed control loop response of barrier drive 2 are considered. Figure 55 thus illustrates evaluation of the HONU-MRAC matrix of dynamics $\bar{\mathbf{M}}$ evaluated about the upper and lower speed limits of the barrier motor, for opening and closing of the barrier boom respectively.

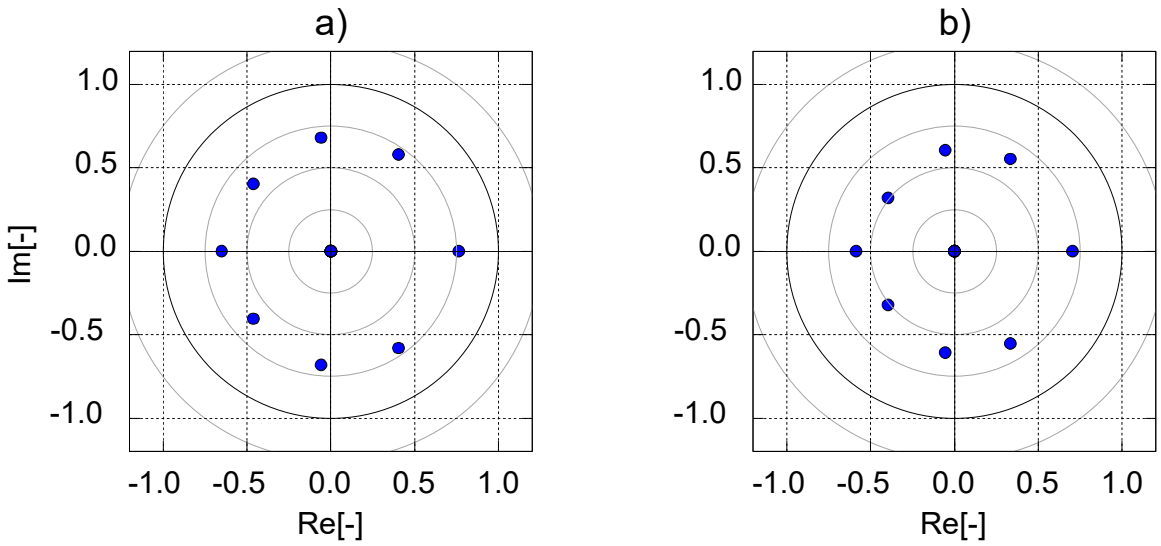


Figure 55 - DHS: Dispersion of HONU model eigenvalues with extended HONU feedback control for real SIM 6/13 Barrier drive control (Figure 32), evaluated for a) Opening b) Closing in region of the upper speed limit. Concept presented in [90], [107] and [96].

From Figure 55 a) and b) representing the eigenvalues of the HONU model with extension of a HONU feedback controller as a closed HONU-MRAC control loop, all eigenvalue positions for control during both opening and closing phases lie with the unit circle and hence, conclude the local BIBO stability of the applied constant parameter HONU-MRAC control loop.

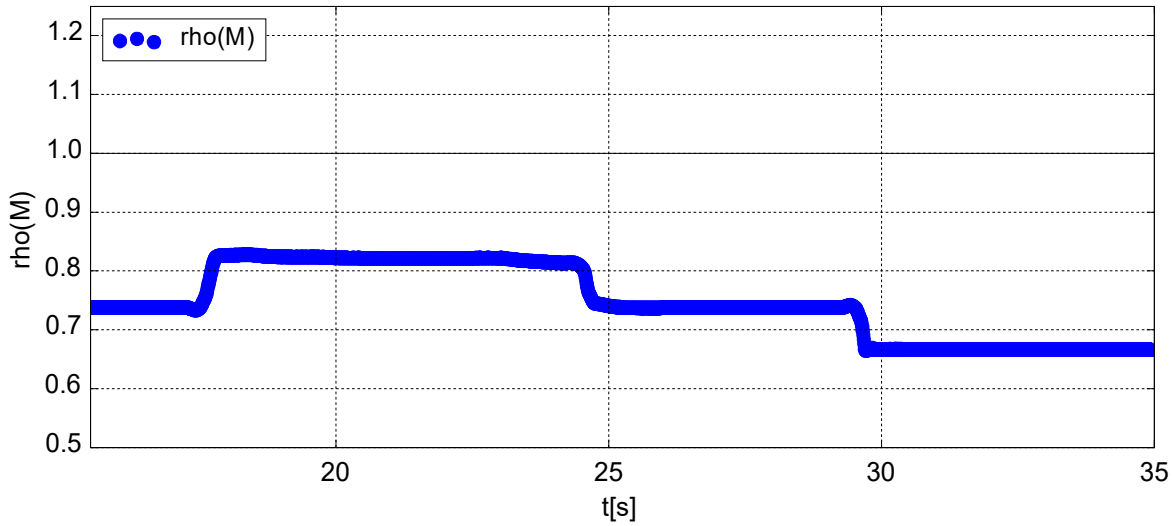


Figure 56: Spectral radii through time rho(M) of HONU model LMD eigenvalues with extended HONU feedback control (Figure 32) on newly tested samples for HONU-MRAC closed control loop. For real control application see section 5.3. Concept presented in [90], [107] and [96].

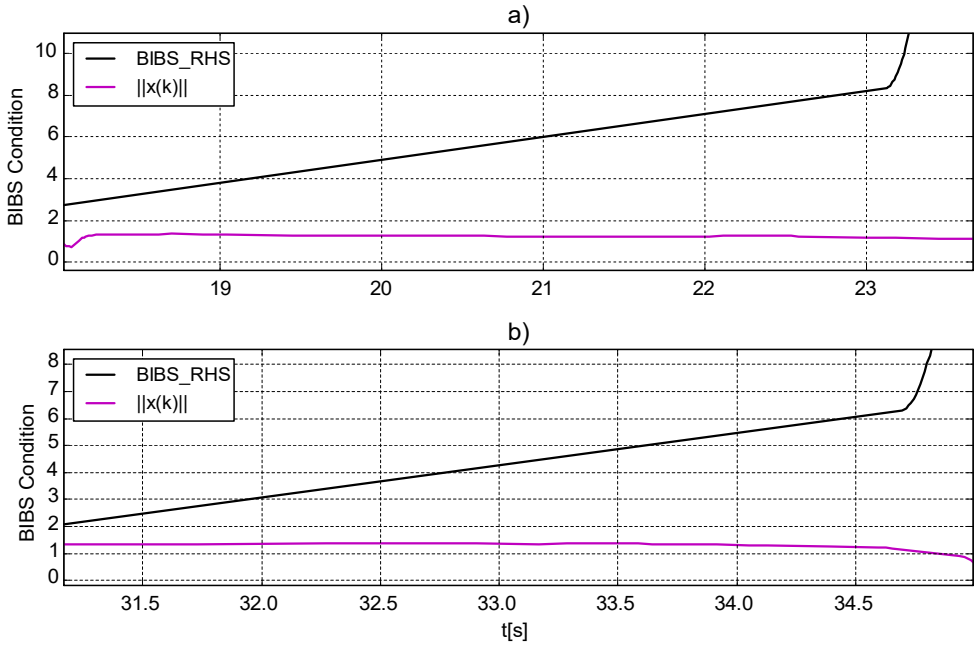


Figure 57- DDHS: Evaluation through time of BIBS condition (134), for Figure 32. a) Barrier upward motion and b) Barrier downward motion in under application of HONU-MRAC control loop with newly tested samples. The right-hand-side (RHS) of (134) is greater than the normalised state vector norm and hence satisfying the condition of BIBS stability. Concept presented in [90], [107] and [96].

Figure 56 illustrates the dispersion of spectral radii for the HONU LMD $\hat{\mathbf{M}}$ in respective operational regions for opening and closing of the barrier drive motion. From the time 17 [s] during opening of the barrier, all spectral radii are positioned well within the unit value, further justified from the local BIBO analysis via DHS. This result is similarly reflected from the time region 29 [s] during the closing phase of the barrier boom arm. From these results, an analysis of the BIBS stability condition in (134) shows the normalised state vector norm is less than the sum of STM matrix norms of the BIBS stability condition (134) (Figure 57), which concludes the HONU-MRAC control loop is also BIBS stable along its trajectory across all state points in the opening and closing regions.

As the applied operation of the barrier drive is fixed for opening and closing regime, a simplification of the BIBS condition (134) can be evaluated for such repeated application.
If the norm of the state transition matrix calculated across the defined operating region is converging to the threshold defined by the value B_1 i.e. $\left\| \prod_{\kappa=k_0}^{k-1} \hat{\mathbf{M}}(\kappa) \right\| \leq B_1$ (defined in Table 9).

Further, that the sum of norms of the STM as a product with the input matrix i.e. $\sum_{\kappa=k_0}^{k-1} \left\| \prod_{i=\kappa}^{k-1} \hat{\mathbf{M}}(i) \cdot \hat{\mathbf{B}}_{\mathbf{a}}(\kappa) \right\| \leq B_2$ converges to the threshold B_2, sufficient constraints are proven

for the HONU-MRAC closed control loop to maintain BIBS stability along its trajectory, with given bounded control inputs.

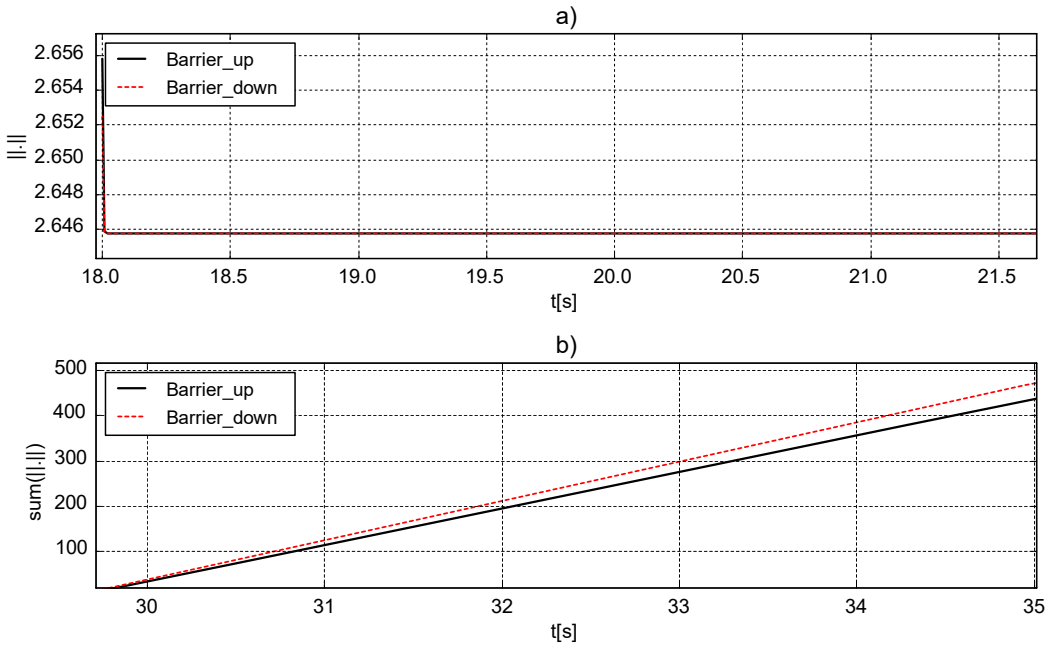


Figure 58 - Evaluation through time of forced and unforced BIBS condition in (134) for Figure 32. a) Barrier upward motion and b) Barrier downward motion in. The respective matrix norms indicate boundary values for convergence of the HONU-MRAC control loop with the respective operating region for upward and downward barrier control. Concept presented in [90], [107] and [96].

TABLE 9

STATE TRANSITION MATRIX (STM) NORM ANALYSIS OF DDHS

Motion Phase	$\left\| \prod_{\kappa=k_0}^{k-1} \hat{\mathbf{M}}(\kappa) \right\| \leq B_1.$	$\sum_{\kappa=k_0}^{k-1} \left\| \prod_{i=\kappa}^{k-1} \hat{\mathbf{M}}(i) \cdot \hat{\mathbf{B}}_{\mathbf{a}}(\kappa) \right\| \leq B_2$
Opening	2.645751	468.4652
Closing	2.645753	504.5006

Following design of a HONU-MRAC closed control loop for the barrier drive control board, the presented results indeed yield superior performance in optimising the existing closed cascade control loop applied previously on the barrier drive system, such to maintain the prescribed motor protection time for overall barrier motion whilst ensuring the desired barrier motor speed is kept maintaining the tolerances of motor speed and current. The derived HONU-MRAC control loop yields to be justified in terms of BIBS stability during application across its respective operational points verified via the DHS method in vicinity of the upper speed limit regions of the barrier drive motion and further complementing the results of the DDHS for monitoring the stability of the applied control loop during transition across respective operational state points. As illustrated in the results of Figure 32 and Figure

33, the real implementation of the HONU-MRAC closed control loop, yields a more universally function control design which can be identified for further loading cases of the barrier arm and further the possibility to extend the problem as a multiple input system with the loading being a further variables to the HONU model parameters. Also the possibilities of online re-tuning during maintenance are highlighted in situations where mechanical differences due to prolonged operation may alter the performance of the tested control output. This result also serves as a promising direction for implementation of a standalone HONU-MRAC control loop for on the up-coming redesign of the barrier drive control board which is planned onwards from the year 2019.

9 Discussion

The first main objective of this dissertation was to extend on the till now developed approaches of HONUs architectures for feedback adaptive control, namely as a new subclass of MRAC control loop design. The extension of the RLS algorithm in this work and deeper experimental and practical analysis of its application to real-time dynamic system control has proven its effectiveness as an incremental supervised leaning algorithm for HONU adaptive identification and further control. Further, as was seen in Figure 38 the application of less appropriate linear (LNU) architecture for identification of the non-linear position feedback system however, using incremental training was still capable to sufficiently identify the contemporary non-linear dynamics when applied in real time. Therefore, in contrast to the GD and L-M algorithm rather than being driven by a minimisation of square error, the RLS algorithm minimizes a weighted cost function with relating to the input signals. Though featuring substantially faster convergence, the algorithm is computationally more complex and should be respected for systems which require rather high sampling intervals. However, in general the presented HONU-MRAC closed control loop approach is strong rather for linear systems with complexities such as oscillatory behaviour and low to moderately non-linear systems (i.e. systems which can be approximated via a recurrent order HONU of $r \leq 3$, unlike the torsional pendulum example in [97] which featured strong sinusoidal non-linearity), which motivate study for alternative control forms like in section 10.2. A key property that lies within the design of any form of conventional or non-conventional form of control is to ensure the stability of the derived and applied control set up for application to the given engineering process. From the reviewed works more traditional approaches feature construction of an adaptive control law which is based on a proven stability condition for any control input. The most readily used being a Lyapunov functional based law; with a further

motive prove the stability about a global system equilibrium point. Following a recent review article of RNN stability methods [48] a natural conclusion was derived that no one method may be universally applicable and that certain methods each have their own advantages for the given control loop design.

Thus, in this dissertation two novel methods were presented. The DHS was derived and mathematically proven via means of Lyapunov matrix inequality relation (96) to justify BIBO stability in neighbourhood of the applied state (operational) point. For practical control problems with varying input the evaluation of local spectral radii (88) can be applied however as seen in Figure 47, due to the relation (135) accounting for previous samples of HONU-MRAC state transitions, DDHS(Strict) yields a stronger condition that clearly pronounces an onset of instability. It further justifies the trajectory in state space for the given control input is becoming unstable as opposed to the local dynamics in vicinity of the discrete state point. However, via analysis of the influence of adaptive gain parameter p as in Figure 44 the evaluation of $\rho(.)$ in (88) can further be used as a tool for correct setting of the magnitude of the feedback gain parameter, for stable process dynamics of the whole HONU-MRAC closed control loop. The DHS method is however limited by polynomial orders due to linearization for a localised state point. Thus, the DDHS was derived and developed as another key contribution of this dissertation work for evaluating the HONU-MRAC closed control loop maintains dynamical stability along its trajectory in state-space. An advantage of this approach is the avoidance of approximation due to building the BIBS stability condition (133) from the fundamental polynomial structure of HONUs themselves as presented in [90] and most recently for HONU-MRAC closed control loops [89] & [97] which makes its application suitable for further polynomial orders.

10 Next Research Directions

10.1 Extension to Other Polynomial Neural Network Based Architectures

A final remark to be stated is the extension of the presented DDHS approach to similar PNN based architectures of standalone polynomial neural based models or whole closed control loops. In certain engineering applications it can also serve purpose to apply the control law (12), or perhaps as further presented in [68], [97], [107] & [99] the extension of multiple HONU feedback controller was further defined as

$$u(k) = p\cdot\left(d(k) - \sum_{\iota=1}^{n_q} q_\iota(k) \right), \tag{152}$$

where n_q in fact denotes the number of single HONU feedback controllers. With this notion, let us consider the derived expressions of the local characteristic coefficients $\hat{\alpha}_i$ for an LNU-QNU closed control loop. Then, on considering the control law (152) in the sense of a dynamic HONU model it yields that

$$\hat{\alpha}_i = w_i - p\cdot\sum_{j=1}^{n_u} w_{n_y+j}\cdot\sum_{l=1}^{n_q} C_i(q_l(k-j)) \quad for \quad i=1,2,3,..,n_y. \tag{153}$$

However, although the scope of this dissertation was focussed to dynamical (recurrent) HONUs, in certain engineering applications we may also consider the need for online adaptive identification via a static HONU, with extension of a dynamical HONU as a feedback controller. In taking regard that the model is a standalone static HONU, then the relation (153) results to be

$$\hat{\alpha}_i = -p\cdot\sum_{j=1}^{n_u} w_{n_y+j}\cdot\sum_{l=1}^{n_q} C_i(q_l(k-j)) \quad for \quad i=1,2,..,n_y+n_u, \tag{154}$$

where the corresponding coefficient terms $\hat{\beta}_i$ for the input vector $\hat{\mathbf{u}}(k-1)$ may then be computed as

$$\hat{\beta}_i = \begin{cases} p\cdot\left[w_{n_y+i} - \sum_{j=1}^{n_u} w_{n_y+j}\cdot\sum_{l=1}^{n_q} C_i(q_l(k-j)) \quad for \quad i=1,..,n_u \right] \\ -p\cdot\sum_{j=1}^{n_u} w_{n_y+j}\cdot\sum_{l=1}^{n_q} C_i(q_l(k-j)) \quad for \quad i=n_u+1,..,n_u+n_u, \end{cases} \tag{155}$$

where the resulting expressions for computation of the canonical state-space form as in (118) then yield to be analogical.

10.2 IRW Active Wheelset Control: A Sliding Mode Control Approach with HONUs

Due to the nature of IRW configuration, another appropriate form of control is to stabilize the lateral skew of the bogie frame to a zero position under applied velocity of the rollers via sliding mode design. Several works in field of discrete-time neural network sliding mode control include [47], [52], [53]. Being inspired from their works, another approach is

proposed here falling from the pointwise HONU representation presented in section 6.1. To develop the approach, let us for comprehensibility consider a one-step ahead QNU predictor, which in this example is investigated as dynamic. However, for highly nonlinear systems can also be considered in static form provided a sufficiently small error margin. From (59), it yields that

$$\overline{x}_{n_y}(k+1)=w_{0,0}+\sum_{j=1}^{n_{\overline{x}}}w_{0,j}\cdot\overline{x}_j(k)+\sum_{i=1}^{n_{\overline{x}}}\sum_{j=i}^{n_{\overline{x}}}w_{i,j}\cdot\overline{x}_i(k), \tag{156}$$

where w holding the same indexing maybe be derived (3) and $\overline{x}_i$ may be further represented generally in long vector form as

$$\overline{\mathbf{x}}(k)=\left[\tilde{y}(k-n_y+1)\quad ..\quad \tilde{y}(k)\quad u(k-n_u+1)\quad ..\quad u(k)\right]^T, \tag{157}$$

where in analogy to (59), applying the affine control formulation with additional integrator input term $\overline{u}(k)$, yields the following set of state equations

$$\begin{aligned}
&\overline{x}_1(k+1)=\tilde{y}(k-n_y+1)\\
&\overline{x}_2(k+1)=\tilde{y}(k-n_y+2)\\
&\vdots\\
&\overline{x}_{n_y}(k+1)=w_{0,0}+\sum_{j=1}^{n_{\overline{x}}}w_{0,j}\cdot\overline{x}_j(k)+\sum_{i=1}^{n_{\overline{x}}}\sum_{j=i}^{n_{\overline{x}}}w_{i,j}\cdot\overline{x}_i(k)\\
&\vdots\\
&\overline{x}_{n_{\overline{x}}}(k+1)=\overline{u}(k).
\end{aligned} \tag{158}$$

Then, to design the sliding mode control let us propose the following switching function

$$\sigma(\overline{\mathbf{x}})=\sum_{i=1}^{n_{\overline{x}}-1}s_i\cdot\overline{x}_i(k)=s_1\cdot\overline{x}_1(k)+s_2\cdot\overline{x}_2(k)+\ldots+s_{n_{\overline{x}}}\cdot\overline{x}_{n_{\overline{x}}}(k);\ \ s_i\in\langle 0,1\rangle, \tag{159}$$

where the variables s_i denote arbitrarily chosen switching weights. Then from (159) we define the sliding mode surface where $\sigma(\overline{\mathbf{x}})=0$. From this, the control objective is for the trajectories in state space to converge to zero, which can be achieved if the system states are forced to push along the sliding surface i.e. $\dot{\sigma}(\overline{\mathbf{x}})=0$. To find a suitable control law, let us consider the following positive-definite Lyapunov function

$$V(\sigma(\overline{\mathbf{x}}))=\frac{1}{2}\sigma(\overline{\mathbf{x}})^T\sigma(\overline{\mathbf{x}}). \tag{160}$$

Then, for asymptotic stability about the origin the Lyapunov function (160)

$$\dot{V}(\sigma(\overline{\mathbf{x}}))=\sigma(\overline{\mathbf{x}})^T\dot{\sigma}(\overline{\mathbf{x}})<0, \tag{161}$$

where from the condition (161)

$$\dot{\sigma}(\overline{\mathbf{x}})=\frac{\partial\sigma(\overline{\mathbf{x}})}{\partial\overline{\mathbf{x}}}\left[\overline{\mathbf{f}}(\overline{\mathbf{x}}(k))+\overline{\mathbf{u}}(k)\right]. \tag{162}$$

In order to maintain the Lyapunov condition then the control law must be chosen so that $\dot{\sigma}<0$ if $\sigma>0$ and $\dot{\sigma}>0$ if $\sigma<0$. Thus, on selecting the switching function and its derivative as

$$\sigma(\overline{\mathbf{x}})=\overline{x}_1(k)+\overline{x}_2(k)+\ldots+\overline{x}_{n_{\overline{x}}}(k);s_i=1\ \forall\ i, \tag{163}$$

it yields

$$\dot{\sigma}(\overline{\mathbf{x}})=\tilde{y}(k-n_y+2)+..+u(k-n_u+2)+..w_{0,0}+\sum_{j=1}^{n_{\overline{x}}}w_{0,j}\cdot\overline{x}_j(k)+\sum_{i=1}^{n_{\overline{x}}}\sum_{j=i}^{n_{\overline{x}}}w_{i,j}\cdot\overline{x}_i(k)+\overline{u}(k)<0\,. \tag{164}$$

Then to maintain (161), we may obtain that if the switching function $\sigma>0$ then

$$\overline{u}(k)<-\left(\tilde{y}(k-n_y+2)+..+u(k-n_u+2)+..w_{0,0}+\sum_{j=1}^{n_{\overline{x}}}w_{0,j}\cdot\overline{x}_j(k)+\sum_{i=1}^{n_{\overline{x}}}\sum_{j=i}^{n_{\overline{x}}}w_{i,j}\cdot\overline{x}_i(k)\right), \tag{165}$$

analogically the condition for $\dot{\sigma}>0$ is then apparent. Then, on defining the following upper limit, which we assume to be a known constant U_k

$$\sup\left(\left|w_{0,0}+\sum_{j=1}^{n_{\overline{x}}}w_{0,j}\cdot\overline{x}_j(k)+\sum_{i=1}^{n_{\overline{x}}}\sum_{j=i}^{n_{\overline{x}}}w_{i,j}\cdot\overline{x}_i(k)\right|\right)\leq U_k\,, \tag{166}$$

applying the rule of triangular inequality from relation (166), and considering the constant $\overline{\varepsilon}\in<0,1>$, then a possible stabilizing input for the system presented in Figure 60 as expressed in the form (158) may be given as

$$\overline{u}(\overline{\mathbf{x}})=\begin{cases}\left|\sum_{i=1}^{n_{\overline{x}}-1}\overline{x}_i(k+1)\right|+U_k+\overline{\varepsilon} & if\ \sigma(\overline{\mathbf{x}})<0;\forall i\neq n_y\\ -\left(\left|\sum_{i=1}^{n_{\overline{x}}-1}\overline{x}_i(k+1)\right|+U_k+\overline{\varepsilon}\right) & if\ \sigma(\overline{\mathbf{x}})>0;\forall i\neq n_y.\end{cases} \tag{167}$$

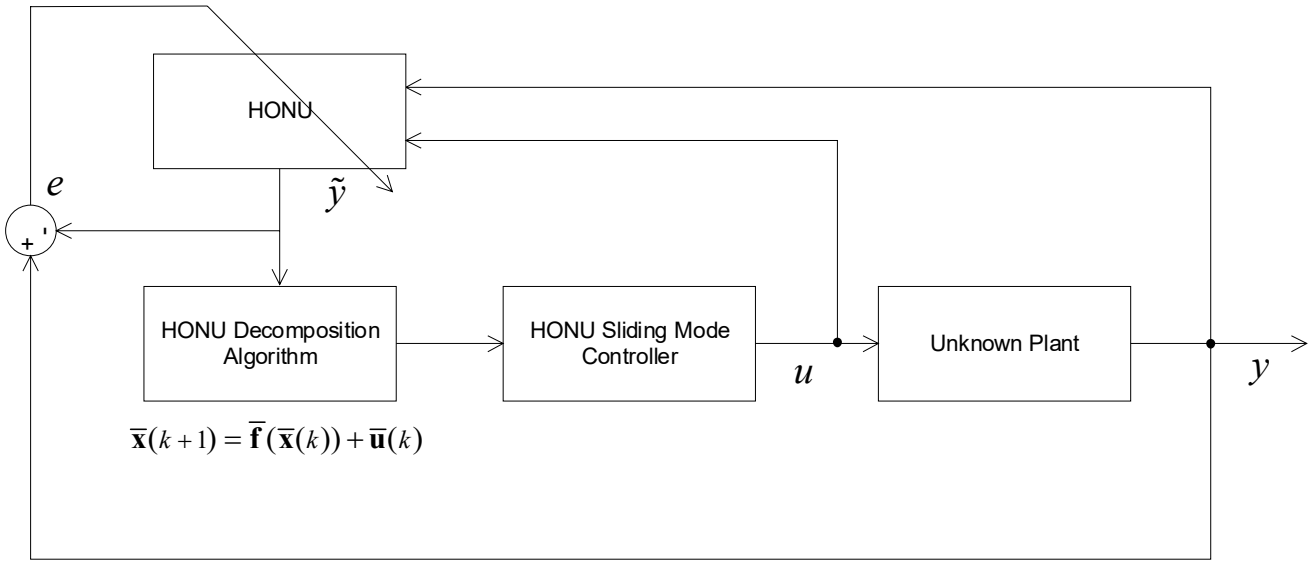


Figure 59 – Block schematic of pointwise state-space HONU approach from (59) to sliding mode control (156)-(167)

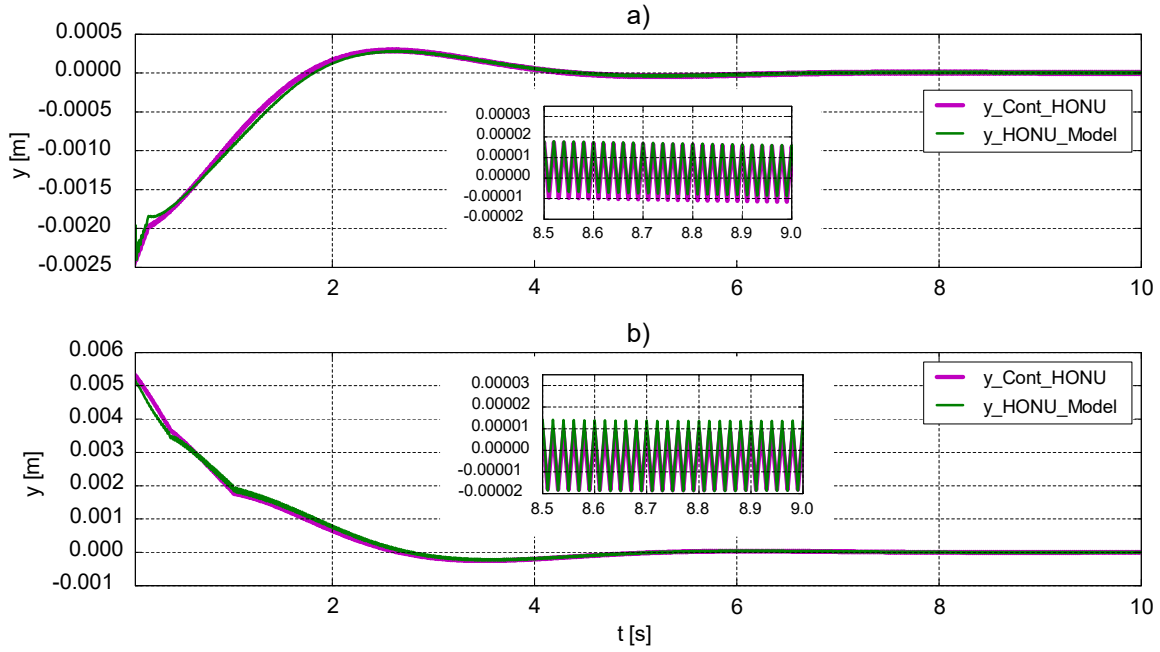


Figure 60 – Sliding mode control via HONU decomposition for IRW wheelset stabilisation of Roller Rig mathematical model (37)-(41). Dynamic QNU model trained in real time via RLS algorithm, where $n_y = 3$, $n_u = 2$ *.a) Plant with HONU sliding mode control* (167) *with roller velocity 50rpm from -2.5mm initial condition b) Plant with HONU sliding mode control* (167) *with roller velocity 50rpm from 5mm initial condition.*

Figure 60 illustrates the performance of the derived sliding mode control law via QNU decomposition. A dynamic QNU featuring 3 previous process outputs n_y and 2 previous inputs trained in real time via the RLS learning algorithm with 0.05[s] of sampling. The results from Figure 60 shows the capability of such HONU architecture to rapidly identify the system dynamics even with nonlinear capabilities and stabilise the IRW configuration of

the mathematical roller rig model from both extremes of initial conditions (5[mm] and -2.5[mm]) and converge to zero within 6[s] of control.

From these initial results in sliding mode control via HONUs, motivation arises for investigation of the HONU decomposition form (108) as an alternative approach. Further, experimentation at increased roller speeds of IRW configuration which remains a challenge for the HONU-MRAC control design. Therefore, for IRW wheelset control a sliding mode design can be a more appropriate as the control objective is focussed more so on system stabilization and is thus a new direction for research of active wheelset control.

11 Conclusion

In this dissertation work, a preliminary objective was to further investigate the applied theory behind HONUs as a rather novel field of MRAC adaptive control. From this work, further investigation behind the advantages of various HONU architectures and their respective supervised learning algorithms was presented in both theoretical as well as real industrial engineering examples. From their respective results, this work may conclude the effectiveness of the approach as a means of optimisation of existing conventional control loops of both linear and nonlinear system dynamics as presented in Figure 24 & Figure 32. Further, an effective form of standalone HONU-MRAC closed loop control as seen in the sense of the two-tank experiment in section 5.1. In addition, a major component of this dissertation work was to provide an extension to the applied theory of HONU adaptive identification and control in the form a newly derived and developed stability analysis approach. From this dissertation two novel methodologies were derived and tested namely the DHS method which proved to be an effective method for concluding BIBO stability of a HONU model or further HONU-MRAC closed control loop for an applied operating region in close neighbourhood or local BIBO of an evaluated state point. As a further means of validation, the DHS method was used to validate the DDHS, inspired from an intrinsic relation of a HONU polynomial equation to that of a reorganised subset of HONU polynomials. From this the sample-by-sample decomposition of the DDHS method proves effective as a means of evaluating the local BIBO stability or further BIBS stability of a HONU-MRAC closed control loop along its dynamical trajectory in state-space, which is further applicable for higher orders of HONU architectures. Following from the proposed relation (134), a further method was derived DDHS(Strict) (135)-(136) which clearly

illustrates an onset of unstable process dynamics due to the fundamental canonical state-space form of a HONU-MRAC control loop as defined in (117)-(118).

Following from the reviewed concepts of discrete-time model and state-of-the-art neural networks stability theories, the DHS and DHS method serves as a practical approach in comparison to other stability approaches in the field of recurrent neural networks which are centred around rather comprehensive mathematical descriptions of all process functions or further their application towards computational demanding control algorithms. Thus, this dissertation provides a foundation into practical stability analysis which can be readily applied for HONU architectures of general orders. Further directions of research were also highlighted via an introduction of sliding mode control via the decomposed HONU method presented in 5.2.2 and extension of the DHS and DDHS approached to other polynomial based neural architectures, like a multi-layered HONU feedback control form as presented in the recent work. As a further outcome of this dissertation the ASPI Kit v 1.3 software [93]-[94] & [98] was released as well as a stability analysis library in the programming language Python software to help engineers and practitioners investigate the potentials of HONU adaptive identification and control for their respective engineering applications.

12 Literature

12.1 List of Cited Publications

[1] A. G. Alexandrov and M. V. Palenov, "Adaptive PID controllers: State of the art and development prospects," *Autom. Remote Control*, vol. 75, no. 2, pp. 188–199, Feb. 2014.

[2] M. Valašek, Z. Šika, T. Vampola, and S. Hecker, "Reduced-Order Modeling," in *Modeling and Control for a Blended Wing Body Aircraft*, Springer International Publishing, 2015.

[3] "Modeling, simulation and control of mechatronical systems," in *Simulation Techniques for Applied Dynamics*, vol. 507, Springer-Verlag Wien, 2009, pp. 75–140.

[4] P. Benes and M. Valašek, "Optimized Re-entry Input Shapers," *J. Theor. Appl. Mech.*, vol. 54, no. 2, pp. 353–368, Warsaw 2016.

[5] M. Valašek, Z. Neusser, P. Neuman, and M. Nečas, "Wave-based Modeling and Control of Interconnected Synchronous Machines - Application on Mechanical Model," *IFAC-Pap.*, vol. 49, no. 27, pp. 352–357, 2016.

[6] M. Valašek and P. Steinbauer, "Nonlinear Control of Multibody Systems," presented at the Euromech, Lisabon, Portugal, 1999, pp. 437–444.

[7] T. Cimen, "State-Dependent Riccati Equation (SDRE) Control: A Survey," in *Proceedings of the 17th World Congress of International Federation of Automatic Control*, Seoul, Korea, 2008.

[8] B. Chen, K. Liu, X. Liu, P. Shi, and C. Lin, "Approximation-based adaptive neural control design for a class of nonlinear systems," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 610–619, May 2014.

[9] H. Wang, K. Liu, X. Liu, B. Chen, and C. Lin, "Neural-based adaptive output-feedback control for a class of nonstrict-feedback stochastic nonlinear systems," *IEEE Trans Cybern*, vol. 45, no. 9, pp. 1977–1987, Sep. 2015.

[10] W. Meng, Q. Yang, J. Si, and Y. Sun, "Adaptive neural control of a class of output-constrained nonaffine systems," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 85–95, Jan. 2016.

[11] S. C. Tong, T. Wang, Y. Li, and H. Zhang, "Adaptive neural network output feedback control for stochastic nonlinear systems with unknown dead-zone and unmodeled dynamics," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 910–921, Jun. 2014.

[12] M. Wang, X. Liu, and P. Shi, "Adaptive neural control of pure-feedback nonlinear time-delay systems via dynamic surface technique," *IEEE Trans. Syst. Man Cybern.*, vol. 41, no. 6, pp. 1681–1692, Dec. 2011.
[13] S. Yin, P. Shi, and H. Yang, "Adaptive fuzzy control of strict-feedback nonlinear time-delay systems with unmodeled dynamics," *IEEE Trans Cybern*, vol. 46, no. 8, pp. 1926–1938, Aug. 2016.
[14] S. Yin, H. Yang, H. Gao, J. Qiu, and O. Kaynak, "An adaptive NN-based approach for fault-tolerant control of nonlinear time-varying delay systems with unmodeled dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*
[15] I. Kanelakopoulos, P. V. Kokotovic, and A. S. Morse, "Systematic design of adaptive controller for feedback linearizable systems," *IEEE Trans. Autom. Control*, vol. 36, no. 11, pp. 1241–1253, Nov. 1991.
[16] X. P. Liu, X. Gu, and K. M. Zhou, "Robust stabilization of MIMO nonlinear systems by back-stepping," *Automatica*, vol. 35, no. 5, pp. 987–992, 1999.
[17] X. J. Xie and J. Tian, "Adaptive state-feedback stabilization of high-order stochastic systems with nonlinear parameterization," *Automatica*, vol. 45, no. 1, pp. 126–133, 2009.
[18] B. Chen, X. P. Liu, K. F. Liu, and C. Lin, "Direct adaptive fuzzy control of nonlinear strict-feedback systems," *Automatica*, vol. 45, no. 6, pp. 1530–1535, 2009.
[19] S. Heidari, A. Shahcheraghi, K. Heidari, S. Zahmatkesh, and F. Piltan, "Intelligent Adaptive Gain Back-stepping Technique," *IJ Inf. Technol. Comput. Sci.*, vol. 7, no. 2, pp. 60–67, 2015.
[20] A. Y. Alanis, E. N. Sanchez, and A. G. Loukianov, "Discrete-Time Adaptive Backstepping Nonlinear Control via High-Order Neural Networks," *IEEE Trans Neural Netw.*, vol. 18, no. 4, pp. 1185–1195, Jul. 2007.
[21] E. N. Sanchez, A. Y. Alanis, and A. G. Loukianov, *Discrete-Time High Order Neural Control - Trained with Kalman Filtering*. Springer-Verlag Berlin Heidelberg, 2008.
[22] S. Kersting and M. Buss, "Direct and Indirect Model Reference Adaptive Control for Multivariable Piecewise Affine Systems," *IEEE Trans Autom Control*, vol. 62, no. 11, pp. 5634–5649, Nov. 2017.
[23] H. Qiang, "The design of model reference adaptive control," presented at the Automatic Control and Artificial Intelligence (ACAI 2012), Xiamen, China, 2012, pp. 2092–2095.
[24] T. H. Liu, Y. C. Lee, and Y. H. Chang, "Adaptive Controller Design for a Linear Motor Control System," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 2, pp. 601–616, 2004.
[25] A. S. Kumar, M. Subba Rao, and Y. S. K. Babu, "Model reference linear adaptive control of DC motor using fuzzy controller," in *TENCON 2008 - 2008 IEEE Region 10 Conference*, 2008, pp. 1–5.
[26] W. K. Lee, C. H. Hyun, H. Lee, E. Kim, and M. Park, "Model Reference Adaptive Synchronization of T-S Fuzzy Discrete Chaotic Systems Using Output Tracking Control," *Chaos Solitons Fractals*, vol. 34, no. 5, pp. 1590–1598, 2007.
[27] C. W. Anderson, D. C. Hittle, A. D. Katz, and M. Kretchmar, "Reinforcement Learning, Neural Networks and PI Control Applied to a Heating Coil," *Artif. Intell. Eng.*, vol. 11, no. 4, pp. 421–429, 1997.
[28] H. D. Patino and D. Liu, "Neural network-based model reference adaptive control system," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 30, no. 1, pp. 198–204, Feb. 2000.
[29] A. N. Chernodub, "Local control gradients criterion for selection of neuroemulators for model reference adaptive neurocontrol," *Opt. Mem. Neural Netw.*, vol. 21, no. 2, pp. 126–131, Jun. 2012.
[30] D. Liu, D. Wang, and D. Zhao, "Adaptive Dynamic Programming for Optimal Control of Unknown Nonlinear Discrete-Time Systems," presented at the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), Paris, 2011.
[31] D. Liu and Q. Wei, "Policy Iteration Adaptive Dynamic Programming Algorithm for Discrete-Time Non-linear Systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.
[32] L. Smetana, "Nonlinear Neuro-Controller for Automatic Control Laboratory System," Master's Thesis, Czech Technical University in Prague, Prague, Czech Republic, 2008.
[33] I. Bukovsky, S. Redlapalli, and M. M. Gupta, "Quadratic and cubic neural units for identification and fast state feedback control of unknown nonlinear dynamic systems," in *Fourth International Symposium on Uncertainty Modeling and Analysis, 2003. ISUMA 2003*, 2003, pp. 330–334.
[34] I. Bukovsky, N. Homma, L. Smetana, R. Rodriguez, M. Mironovova, and S. Vrana, "Quadratic neural unit is a good compromise between linear models and neural networks for industrial applications," in *2010 9th IEEE International Conference on Cognitive Informatics (ICCI)*, 2010, pp. 556–560.
[35] G. P. Liu, V. Kadirkamanathan, and S. A. Billings, "On-line identification of nonlinear systems using Volterra polynomial basis function neural networks," *Neural Netw.*, vol. 11, no. 9, pp. 1645–1657, Dec. 1998.
[36] M. M. Gupta, I. Bukovsky, N. Homma, A.M. G. Solo, and Z.-G.. Hou, "Fundamentals of Higher Order Neural Networks for Modeling and Simulation," in *Artificial Higher Order Neural Networks for Modeling and Simulation*, Ming Zhang, Ed. Hershey, PA, USA: IGI Global, 2013, pp. 103–133.
[37] B. Xu and F. Sun, "Composite Intelligent Learning Control of Strict-Feedback Systems With Disturbance," *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–12, Jan. 2017.
[38] B. Xu, Z. Shi, C. Yang, and F. Sun, "Composite Neural Dynamic Surface Control of a Class of Uncertain Nonlinear Systems in Strict-Feedback Form," *Dec 2014*, vol. 44, no. 12, pp. 2626–2634.

[39] I. Bukovsky and N. Homma, “An Approach to Stable Gradient Descent Adaptation of Higher-Order Neural Units,” *IEEE Trans. Neural Netw. Learn. Syst.*, 2016.
[40] J. Fei and H. Ding, “Adaptive sliding mode control of dynamic system using RBF neural network,” *Nonlinear Dyn. 70*, pp. 1563–1573, Aug. 2012.
[41] N. E. Barabanov and D. V. Prokhorov, “A New Method for Stability Analysis of Nonlinear Discrete-Time Systems,” *IEEE Trans. Autom. Control*, vol. 48, no. 12, pp. 2250–2255, Dec. 2003.
[42] X. Liao, G. Chen, and E. N. Sanchez, “LMI-based approach for asymptotically stability analysis of delayed neural networks,” *IEEE Trans. Circuits Syst. Fundam. Theory Appl.*, vol. 49, no. 7, pp. 1033–1039, Jul. 2002.
[43] X. Liao, G. Chen, and E. N. Sanchez, “Delay-dependent exponential stability analysis of delayed neural networks: an LMI approach,” *Neural Netw.*, vol. 16, no. 10, pp. 1401–1402, Dec. 2003.
[44] A. Bacciotti and A. Biglio, “Some remarks about stability of nonlinear discrete-time control systems,” *Nonlinear Differ. Equ. Appl.*, vol. 8, no. 4, pp. 425–438, Apr. 2000.
[45] E. D. Sontag, “Input to State Stability: Basic Concepts and Results,” in *Nonlinear and Optimal Control Theory*, vol. 1932, Springer Berlin Heidelberg, 2008, pp. 163–220.
[46] S. Zhu and Y. Shen, “Two algebraic criteria for input-to-state stability of recurrent neural networks with time-varying delays,” *Neural Comput Applic*, vol. 22, no. 6, pp. 1163–1169, May 2013.
[47] W. Yu, “Neural Feedback Passivity of Unknow Nonlinear Systems via Sliding Mode Technique,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1560–1566, Jul. 2015.
[48] H. Zhang, Z. Wang, and D. Liu, “A Comprehensive Review of Stability Analysis of Continuous-Time Recurrent Neural Networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1229–1262, Jul. 2014.
[49] P. D. Roberts, “A brief overview of model predictive control,” in *Practical Experiences with Predictive Control (Ref. No. 2000/023)*, Middlesbrough, UK, 2000.
[50] O. Šantin and V. Havlena, “Gradient projection based algorithm for large scale real time model predictive control,” in *2011 Chinese Control and Decision Conference (CCDC)*, Mianyang, China, 2011.
[51] O. Šantin and V. Havlena, “Combined partial conjugate gradient and gradient projection solver for MPC,” in *2011 IEEE International Conference on Control Applications (CCA)*, Denver, CO, USA, 2011.
[52] J. Fei, “Adaptive Sliding Mode Control of Dynamic Systems Using Double Loop Recurrent Neural Network Structure,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1275–1286, Apr. 2018.
[53] M. Elena Antonio-Toledo, P. Ramirez, E. N. Sanchez, and A. G. Loukianov, “Discrete-Time Neural Block Control Using Sliding Modes for Induction Motors with Gears,” *Proc. 19th World Congr. Int. Fed. Autom. Control IFAC*, vol. 47, no. 3, Aug. 2014.
[54] F. Y. Wang, H. Zhang, and D. Liu, “Adaptive Dynamic Programming: An Introduction,” *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 39–47, May-2009.
[55] L. Busoniu, D. Ernst, B. D. Schutter, and R. Babuska, “Approximate reinforcement learning: An overview,” in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, Paris, France, 2011.
[56] K. B. Pathak and D. M. Adhyaru, “Survey of Model Reference Adaptive Control,” in *2012 Nirma University International Conference on Engineering (NUiCONE)*, Ahmedabad, India, 2012.
[57] Y. Yang, C. Zhou, and J. Ren, “Model reference adaptive robust fuzzy control for ship steering autopilot with uncertain nonlinear systems,” *Appl. Soft Comput.*, vol. 3, no. 4, pp. 305–316, Dec. 2003.
[58] X. Wu, X. Wu, Q. Luo, and X. Guan, “Neural network-based adaptive tracking control for nonlinearly parameterized systems with unknown input nonlinearities,” *Neurocomputing*, vol. 82, pp. 127–142, Apr. 2012.
[59] A. N. Chernodub, “Local control gradients criterion for selection of neuroemulators for model reference adaptive neurocontrol,” *Opt. Mem. Neural Netw.*, vol. 21, no. 2, pp. 126–131, Jun. 2012.
[60] S. C. Tong, X. L. He, and H. G. Zhang, “A combined back-stepping and small-gain approach to robust adaptive fuzzy output feedback control,” *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 5, pp. 1059–1069, Oct. 2009.
[61] T. Zhang, S. S. Ge, and C. C. Hange, “Adaptive neural network control for strict-feedback non-linear systems using back-stepping design,” *Automatica*, vol. 36, no. 12, pp. 1835–1846, 2000.
[62] Z. Liu, F. Wang, Y. Zhang, X. Chen, and C. L. P. Chen, “Adaptive Fuzzy Output-Feedback Controller Design for Non-linear Systems via Back-stepping and Small Gain Approach,” *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1714–1725, Oct. 2014.
[63] O. Kovalenko, D. Liu, and H. Javaherian, “Neural network modeling and adaptive critic control of automotive fuel-injection systems,” in *Proceedings of the 2004 IEEE International Symposium on Intelligent Control, 2004*, 2004, pp. 368–173.
[64] S. Formentin, K. V. Heusden, and A. Karami, “A Comparison of Model-Based and Data-Driven Controller Tuning,” *Int. J. Adapt. Control Signal Process.*, vol. 28, no. 10, pp. 882–897, 2014.
[65] M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*. John Wiley & Sons, 2004.

[66] J. G. Taylor and S. Coombes, "Learning higher order correlations," *Neural Netw.*, vol. 6, no. 3, pp. 423–427, 1993.
[67] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
[68] I. Bukovsky, J. Voracek, K. Ichiji, and H. Noriyasu, "Higher Order Neural Units for Efficient Adaptive Control of Weakly Nonlinear Systems:," 2017, pp. 149–157.
[69] T. Chu, "An exponential convergence estimate for analog neural networks with delay," *Phys. Lett. A*, vol. 283, no. 1–2, pp. 113–118, May 2001.
[70] Z. Zhang, "Global exponential stability and periodic solutions of delay Hopfield neural networks," *Int. J. Syst. Sci.*, vol. 27, no. 2, pp. 227–231, Feb. 1996.
[71] W. Zhao and Q. Zhu, "New results of global robust exponential stability of neural networks with delays," *Nonlinear Anal. Real World Appl.*, vol. 11, no. 2, pp. 1190–1197, Apr. 2010.
[72] Z.-G. Wu, J. Lam, H. Su, and J. Chu, "Stability and Dissipativity Analysis of Static Neural Networks With Time Delay," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 199–210, Feb. 2012.
[73] S. Arik, "Global asymptotic stability of a class of dynamical neural networks," *IEEE Trans. Circuits Syst. Fundam. Theory Appl.*, vol. 47, no. 4, pp. 568–571, Apr. 2000.
[74] Y. Yasuda, H. Ito, and H. Ishizu, "On the input-output stability of linear time-varying systems," *SIAM J Appl Math*, vol. 38, no. 1, pp. 175–188, Feb. 1980.
[75] Z. Li, Z. Fei, and H. Gao, "Stability and stabilisation of Markovian jump systems with time-varying delay: An input-output approach," *IET Control Theory Appl*, vol. 6, no. 17, pp. 2601–2610, Nov. 2012.
[76] A. Angeli, E. D. Sontag, and Y. Wang, "A characterization of integral input-to-state stability," *IEEE Trans. Autom. Control*, vol. 45, no. 6, pp. 1082–1097, Jun. 2000.
[77] M. Lazar, W. P. M. H. Heemals, and A. R. Teel, "Further input-to-state stability subtleties for discrete-time systems," *IEEE Trans Autom Control*, vol. 58, no. 6, pp. 1609–1613, Jun. 2013.
[78] Z. P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 6, pp. 857–869.
[79] Z. Yang, W. Zhou, and T. Huang, "Exponential input-to-state stability of recurrent neural networks with multiple time-varying delays," *Cogn Neurodyn*, vol. 8, no. 1, pp. 47–54, Feb. 2014.
[80] I. Bukovsky, Z.-G. Hou, J. Bila, and M. M. Gupta, "Foundation of Notation and Classification of Nonconventional Static and Dynamic Neural Units," in *6th IEEE International Conference on Cognitive Informatics*, 2007, pp. 401–407.
[81] J. Kalivoda and P. Bauer, "Design of Experimental Mechatronic Bogie," in *International Symposium on Seed-up and Service Technology for Railway and Maglev Systems: STECH. Chiba*, Tokyo, Japan, 2015.
[82] J. Kalivoda and P. Bauer, "Mechatronic Bogie for Roller Rig Tests," in *IAVSD 2015 - 24th International Symposium on Dynamics of Vehicles on Roads and Tracks*, Graz, Austria, 2015.
[83] S. Myamlin, J. Kalivoda, and L. Neduzha, "Testing of Railway Vehicles Using Roller Rigs," *Procedia Eng.*, vol. 187(2017), pp. 688–695, 2017.
[84] G. F. Franklin, D. J Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 6th ed. Upper Saddle River, N.J: Pearson Prentice Hall, 2010.
[85] Z. Wang and D. Liu, "Stability Analysis for a Class of Systems: From Model-Based Methods to Data-Driven Methods," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6463–6471.
[86] A. Krhovják, P. Dostál, S. Talaš, and L. Rušar, "Multivariable gain scheduled control of two funnel liquid tanks in series," in *2015 20th International Conference on Process Control (PC)*, 2015, pp. 60–65.

12.2 List of All Publications by the Author

[87] P. Benes and I. Bukovsky, "Neural network approach to hoist deceleration control," in *Neural Networks (IJCNN), 2014 International Joint Conference on*, 2014, pp. 1864–1869.
[88] P. M. Benes, I. Bukovsky, M. Cejnek, and J. Kalivoda, "Neural Network Approach to Railway Stand Lateral Skew Control," in *Computer Science & Information Technology (CS& IT)*, Sydney, Australia, 2014, vol. 4, pp. 327–339.
[89] P. Benes and I. Bukovsky, "An Input to State Stability Approach for Evaluation of Nonlinear Control Loops with Linear Plant Model," in *Cybernetics and Algorithms in Intelligent Systems*, vol. 765, R. Silhavy, R. Senkerik, Z. K. Oplatkova, Z. Prokopova, and P. Silhavy, Eds. Springer International Publishing, 2018, pp. 144–154.
[90] P. Benes and I. Bukovsky, "On the Intrinsic Relation between Linear Dynamical Systems and Higher Order Neural Units," in *Intelligent Systems in Cybernetics and Automation Theory*, R. Silhavy, R. Senkerik, Z. K. Oplatkova, Z. Prokopova, and P. Silhavy, Eds. Springer International Publishing, 2016.

[91] P. M. Benes, M. Erben, M. Vesely, O. Liska, and I. Bukovsky, "HONU and Supervised Learning Algorithms in Adaptive Feedback Control," in *Applied Artificial Higher Order Neural Networks for Control and Recognition*, IGI Global, 2016, p. pp.35-60.

[92] I. Bukovsky, P. Benes, and M. Slama, "Laboratory Systems Control with Adaptively Tuned Higher Order Neural Units," in Intelligent Systems in Cybernetics and Automation Theory, R. Silhavy, R. Senkerik, Z. K. Oplatkova, Z. Prokopova, and P. Silhavy, Eds. Springer International Publishing, 2015, pp. 275–284.

[93] P. Benes, "Software Application for Adaptive Identification and Controller Tuning," (Supervisor Ivo Bukovsky) Student's Conference STC, Faculty of Mechanical Engineering, CTU in Prague, April 2013 (received 1st in section prize & Bosch award).

[94] P. Benes, "Software Application for Adaptive Identification and Controller Tuning," (Supervisor Ivo Bukovsky) Student's Conference STOC, Faculty of Applied Informatics, University of Tomas Bata in Zlin, April 2013.

[95] P. Benes, I. Bukovsky, and J. Kalivoda, "Achievements in Neural Network Approach to Railway Stand Lateral Skew Control," presented at the Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2014, Herbertov, Czech Republic, 2014.

[96] P. M. Benes and J. Vojna, "Simulation Analysis of Static and Dynamic Wind Loads for Wayguard DLX Design," presented at the Siemens Simulations Conference (SSC 2017), Siemens Conference Center, Munich Perlach, December 5-6th.

[97] P. M. Benes, I. Bukovsky, M. Vesely, K. Ichiji, and N. Homma, "Framework for Discrete-Time Model Reference Adaptive Control of Weakly Nonlinear Systems with HONUs," chapter in Computational Intelligence, International Joint Conference, IJCCI 2017 Madeira, Portugal, November 1-3, 2017 Revised Selected Papers, J. Merelo et al. Eds. Springer International Publishing. (Published in proceeding chapter).

[98] P. M. Benes, Adaptive Signal Processing and Informatics (ASPI) Kit. Adaptive Signal Processing and Informatics Computational Centre (ASPICC), 2015.

[99] I. Bukovsky, P.M. Benes, M. Vesely, J. Pitel, M. M. Gupta, "Model Reference Multiple-Degree-of-Freedom Adaptive Control with HONUs", The 2016 International Joint Conference on Neural Networks (IJCNN 2016), IEEE WCCI 2016, Vancouver, 2016.

[100] M. Cejnek, P. M. Benes, and I. Bukovsky, "Another Adaptive Approach to Novelty Detection in Time Series," in Computer Science & Information Technology (CS& IT), Sydney, NSW, Australia, AIRCC, 2014, pp. 341–351.

[101] I. Bukovsky, P. M. Benes, C. Oswald, M.Cejnek:" Learning Entropy for Novelty Detection: A Cognitive Approach for Adaptive Filters", Sensor Signal Processing for Defence (SSPD) Conference 2014 , Edinburgh, UK, Sept. 8-9, 2014.

[102] I. Bukovsky. N. Homma, K. Ichiji, M. Cejnek, P.M. Benes & J. Bila "A Fast Neural Network Approach to Predict Lung Tumor Motion during Respiration for Radiation Therapy Applications", BioMed Research International, issue Radiation Oncology and Medical Physics (ROMP), vol. 2015, Article ID 489679, 13 pages, 2015. doi:10.1155/2015/489679. (Submission presented in journal)

[103] I. Bukovsky, M.Vesely, P.M.Benes: "Poznatky z vyzkumu neuro-regulatoru a z laboratorni praxe" in 12th Annual Technical Conference ARAP 2016, Czech Technical University in Prague, Novemeber, 2016, Prague, Czech Republic

[104] V. Maly, M. Vesely, P. Benes, P. Neuman, and I. Bukovsky, "Study of Closed-Loop Model Reference Adaptive Control of Smart MicroGrid with QNU and Recurrent Learning," Acta Mech. Slovaca, vol. 21, no. 4, pp. 34–39, 2017 (Submission presented in journal)

[105] I. Bukovsky, C. Oswald, P.M. Benes, M. Vesely, O. Liska, K. Ichiji, N. Homma, "Z výzkumu adaptivních a optimalizačních metod v identifikaci, řízení a detekci," in Artep 2017, Slovensko, 2017.

[106] P. Benes, I. Bukovsky, and O. Budik, "Striktní stabilita adaptivních dynamických polynomiálních systémů," in Artep 2019, Slovensko, 2019, pp. 26-1–26–17.

[107] I. Bukovsky, P. M. Benes, and M. Vesely, "Introduction and Application Aspects of Machine Learning for Model Reference Adaptive Control With Polynomial Neurons," in Artificial Intelligence and Machine Learning Applications in Civil, Mechanical, and Industrial Engineering, Hershey, PA 17033-1240, USA: IGI Global 2019, pp. 58–84. (Submission presented in book chapter).

13 Appendix

13.1 Example - Pointwise State-Space Representation: Dynamic Nonlinear Neural Unit Model

QNU Nonlinear Model, $n_y = 3$ *and* $n_u = 2$*, state equations*

$$\overline{x}_1(k+1) = \overline{x}_2(k) \tag{168}$$

$$\overline{x}_2(k+1) = \overline{x}_3(k) \tag{169}$$

$$\begin{aligned}\overline{x}_3(k+1) &= w_{0,0} + w_{0,1}\overline{x}_3(k) + w_{0,2}\overline{x}_2(k) + w_{0,3}\overline{x}_1(k) + w_{0,4}\overline{x}_5(k) + \\ &w_{0,5}\overline{x}_4(k) + w_{1,1}\overline{x}_3(k)\overline{x}_3(k) + w_{1,2}\overline{x}_3(k)\overline{x}_2(k) + w_{1,3}\overline{x}_3(k)\overline{x}_1(k) + w_{1,4}\overline{x}_3(k)\overline{x}_5(k) \\ &+ w_{1,5}\overline{x}_3(k)\overline{x}_4(k) + w_{2,2}\overline{x}_2(k)\overline{x}_2(k) + w_{2,3}\overline{x}_2(k)\overline{x}_1(k) + w_{2,4}\overline{x}_2(k)\overline{x}_5(k) \\ &+ w_{2,5}\overline{x}_2(k)\overline{x}_4(k) + w_{3,3}\overline{x}_1(k)\overline{x}_1(k) + w_{3,4}\overline{x}_1(k)\overline{x}_5(k) + w_{3,5}\overline{x}_1(k)\overline{x}_4(k) \\ &+ w_{4,4}\overline{x}_5(k)\overline{x}_5(k) + w_{4,5}\overline{x}_5(k)\overline{x}_4(k) + w_{5,5}\overline{x}_4(k)\overline{x}_4(k)\end{aligned} \tag{170}$$

$$\overline{x}_4(k+1) = u(k) \tag{171}$$

$$\overline{x}_5(k+1) = \overline{u}(k). \tag{172}$$

From relations (168)-(172), the respective state equations may be reformulated to a state-space representation in the following manner

$$\begin{bmatrix}\overline{x}_1(k+1)\\ \overline{x}_2(k+1)\\ \overline{x}_3(k+1)\\ \overline{x}_4(k+1)\\ \overline{x}_5(k+1)\end{bmatrix} = \begin{bmatrix}\overline{x}_2(k)\\ \overline{x}_3(k)\\ w_{0,0} + w_{0,1}\overline{x}_3(k) + w_{0,2}\overline{x}_2(k) + w_{0,3}\overline{x}_1(k) + w_{0,4}\overline{x}_5(k) + \\ w_{0,5}\overline{x}_4(k) + w_{1,1}\overline{x}_3(k)\overline{x}_3(k) + w_{1,2}\overline{x}_3(k)\overline{x}_2(k) + w_{1,3}\overline{x}_3(k)\overline{x}_1(k) \\ +w_{1,4}\overline{x}_3(k)\overline{x}_5(k) + w_{1,5}\overline{x}_3(k)\overline{x}_4(k) + w_{2,2}\overline{x}_2(k)\overline{x}_2(k) + \\ w_{2,3}\overline{x}_2(k)\overline{x}_1(k) + w_{2,4}\overline{x}_2(k)\overline{x}_5(k) + w_{2,5}\overline{x}_2(k)\overline{x}_4(k) + w_{3,3}\overline{x}_1(k)\overline{x}_1(k) \\ +w_{3,4}\overline{x}_1(k)\overline{x}_5(k) + w_{3,5}\overline{x}_1(k)\overline{x}_4(k) + w_{4,4}\overline{x}_5(k)\overline{x}_5(k) + w_{4,5}\overline{x}_5(k)\overline{x}_4(k) \\ +w_{5,5}\overline{x}_4(k)\overline{x}_4(k)\\ \overline{x}_5(k)\\ 0\end{bmatrix} + \begin{bmatrix}0\\0\\0\\0\\ \vdots \\ \overline{u}(k)\end{bmatrix}, \tag{173}$$

$$\overline{y}(k) = \overline{x}_3(k). \tag{174}$$

The coefficients $\overline{a}_{i,j}$ in this example are precisely derived as

$$\overline{a}_{3,1} = w_{0,3} + w_{1,3}\overline{x}_3(k) + w_{2,3}\overline{x}_2(k) + 2w_{3,3}\overline{x}_1(k) + w_{3,4}\overline{x}_5(k) + w_{3,5}\overline{x}_4(k) \tag{175}$$

$$\overline{a}_{3,2} = w_{0,2} + w_{1,2}\overline{x}_3(k) + 2w_{2,2}\overline{x}_2(k) + w_{2,3}\overline{x}_1(k) + w_{2,4}\overline{x}_5(k) + w_{2,5}\overline{x}_4(k) \tag{176}$$

$$\bar{a}_{3,3} = w_{0,1} + 2w_{1,1}\bar{x}_3(k) + w_{1,2}\bar{x}_2(k) + w_{1,3}\bar{x}_1(k) + w_{1,4}\bar{x}_5(k) + w_{1,5}\bar{x}_4(k) \tag{177}$$

$$\bar{a}_{3,4} = w_{0,5} + w_{1,5}\bar{x}_3(k) + w_{2,5}\bar{x}_2(k) + w_{3,5}\bar{x}_1(k) + w_{4,5}\bar{x}_5(k) + 2w_{5,5}\bar{x}_4(k) \tag{178}$$

$$\bar{a}_{3,5} = w_{0,4} + w_{1,4}\bar{x}_3(k) + w_{2,4}\bar{x}_2(k) + w_{3,4}\bar{x}_1(k) + 2w_{4,4}\bar{x}_5(k) + w_{4,5}\bar{x}_4(k). \tag{179}$$

13.2 Example - Pointwise State-Space Representation: Dynamic LNU with HONU Feedback Control

LNU with LNU Feedback controller, $n_y = 4$ *and* $n_u = 3$

$$\bar{\alpha}_{n_y+n_u,1} = -w_7 \cdot p \cdot v_4 \tag{180}$$

$$\bar{\alpha}_{n_y+n_u,2} = -p \cdot [w_6 v_4 + w_7 v_3] \tag{181}$$

$$\bar{\alpha}_{n_y+n_u,3} = -p \cdot [w_5 v_4 + w_6 v_3 + w_7 v_2] \tag{182}$$

$$\bar{\alpha}_{n_y+n_u,4} = w_4 - p \cdot [w_5 v_3 + w_6 v_2 + w_7 v_1] \tag{183}$$

$$\bar{\alpha}_{n_y+n_u,5} = w_3 - p \cdot [w_5 v_2 + w_6 \cdot v_1] \tag{184}$$

$$\bar{\alpha}_{n_y+n_u,6} = w_2 - w_5 \cdot p \cdot v_1 \tag{185}$$

$$\bar{\alpha}_{n_y+n_u,7} = w_1 \tag{186}$$

$$\bar{\alpha}_{n_y+n_u,8} = -w_7 \cdot p \cdot v_7 \tag{187}$$

$$\bar{\alpha}_{n_y+n_u,9} = -p \cdot [w_6 \cdot v_7 + w_7 \cdot v_6] \tag{188}$$

$$\bar{\alpha}_{n_y+n_u,10} = -p \cdot [w_5 \cdot v_7 + w_6 \cdot v_6 + w_7 \cdot v_5] \tag{189}$$

$$\bar{\alpha}_{n_y+n_u,11} = w_7 - p \cdot [w_5 \cdot v_6 + w_6 \cdot v_5] \tag{190}$$

$$\bar{\alpha}_{n_y+n_u,12} = -w_5 \cdot p \cdot v_5 + w_6 \tag{191}$$

$$\bar{\alpha}_{n_y+n_u,13} = w_5. \tag{192}$$

LNU with QNU Feedback controller, $n_y = 4$ *and* $n_u = 3$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,1} &= -w_7 \cdot p \cdot [v_{0,4} + v_{1,4}\bar{x}_4(k) + v_{2,4}\bar{x}_3(k) + v_{3,4}\bar{x}_2(k) + 2v_{4,4}\bar{x}_1(k) \\ &+ v_{4,5}\bar{x}_{10}(k) + v_{4,6}\bar{x}_9(k) + v_{4,7}\bar{x}_8(k)]\end{aligned} \tag{193}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,2} &= -w_6 \cdot p \cdot [v_{0,4} + v_{1,4}\bar{x}_5(k) + v_{2,4}\bar{x}_4(k) + v_{3,4}\bar{x}_3(k) + 2v_{4,4}\bar{x}_2(k) \\ &+ v_{4,5}\bar{x}_{11}(k) + v_{4,6}\bar{x}_{10}(k) + v_{4,7}\bar{x}_9(k)] - w_7 \cdot p \cdot [v_{0,3} + v_{1,3}\bar{x}_4(k) + v_{2,3}\bar{x}_3(k) \\ &+ 2v_{3,3}\bar{x}_2(k) + v_{3,4}\bar{x}_1(k) + v_{3,5}\bar{x}_{10}(k) + v_{3,6}\bar{x}_9(k) + v_{3,7}\bar{x}_8(k)]\end{aligned} \tag{194}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,3} &= -w_5 \cdot p \cdot [v_{0,4} + v_{1,4}\bar{x}_6(k) + v_{2,4}\bar{x}_5(k) + v_{3,4}\bar{x}_4(k) + 2v_{4,4}\bar{x}_3(k) \\ &+ v_{4,5}\bar{x}_{12}(k) + v_{4,6}\bar{x}_{11}(k) + v_{4,7}\bar{x}_{10}(k)] - w_6 \cdot p \cdot [v_{0,3} + v_{1,3}\bar{x}_5(k) + v_{2,3}\bar{x}_4(k) \\ &+ 2v_{3,3}\bar{x}_3(k) + v_{3,4}\bar{x}_2(k) + v_{3,5}\bar{x}_{11}(k) + v_{3,6}\bar{x}_{10}(k) + v_{3,7}\bar{x}_9(k)] \\ &- w_7 \cdot p \cdot [v_{0,2} + v_{1,2}\bar{x}_4(k) + 2v_{2,2}\bar{x}_3(k) + v_{2,3}\bar{x}_2(k) + v_{2,4}\bar{x}_1(k) + v_{2,5}\bar{x}_{10}(k) \\ &+ v_{2,6}\bar{x}_9(k) + v_{2,7}\bar{x}_8(k)]\end{aligned} \tag{195}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,4} &= w_4 - w_5 \cdot p \cdot [v_{0,3} + v_{1,3}\bar{x}_6(k) + v_{2,3}\bar{x}_5(k) + 2v_{3,3}\bar{x}_4(k) + v_{3,4}\bar{x}_3(k) \\ &+ v_{3,5}\bar{x}_{12}(k) + v_{3,6}\bar{x}_{11}(k) + v_{3,7}\bar{x}_{10}(k)] - w_6 \cdot p \cdot [v_{0,2} + v_{1,2}\bar{x}_5(k) + 2v_{2,2}\bar{x}_4(k) \\ &+ v_{2,3}\bar{x}_3(k) + v_{2,4}\bar{x}_2(k) + v_{2,5}\bar{x}_{11}(k) + v_{2,6}\bar{x}_{10}(k) + v_{2,7}\bar{x}_9(k)] \\ &- w_7 \cdot p \cdot [v_{0,1} + 2v_{1,1}\bar{x}_4(k) + v_{1,2}\bar{x}_3(k) + v_{1,3}\bar{x}_2(k) + v_{1,4}\bar{x}_1(k) + v_{1,5}\bar{x}_{10}(k) \\ &+ v_{1,6}\bar{x}_9(k) + v_{1,7}\bar{x}_8(k)]\end{aligned} \tag{196}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,5} &= w_3 - w_5 \cdot p \cdot [v_{0,2} + v_{1,2}\bar{x}_6(k) + 2v_{2,2}\bar{x}_5(k) + v_{2,3}\bar{x}_4(k) + v_{2,4}\bar{x}_3(k) \\ &+ v_{2,5}\bar{x}_{12}(k) + v_{2,6}\bar{x}_{11}(k) + v_{2,7}\bar{x}_{10}(k)] - w_6 \cdot p \cdot [v_{0,1} + 2v_{1,1}\bar{x}_5(k) + v_{1,2}\bar{x}_4(k) \\ &+ v_{1,3}\bar{x}_3(k) + v_{1,4}\bar{x}_2(k) + v_{1,5}\bar{x}_{11}(k) + v_{1,6}\bar{x}_{10}(k) + v_{1,7}\bar{x}_9(k)]\end{aligned} \tag{197}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,6} &= w_2 - w_5 \cdot p \cdot [v_{0,1} + 2v_{1,1}\bar{x}_6(k) + 2v_{1,2}\bar{x}_5(k) + v_{1,3}\bar{x}_4(k) + v_{1,4}\bar{x}_3(k) \\ &+ v_{1,5}\bar{x}_{12}(k) + v_{1,6}\bar{x}_{11}(k) + v_{1,7}\bar{x}_{10}(k)]\end{aligned} \tag{198}$$

$$\bar{\alpha}_{n_y+n_u,7} = w_1 \tag{199}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,8} &= -w_7 \cdot p \cdot [v_{0,7} + v_{1,7}\bar{x}_4(k) + v_{2,7}\bar{x}_3(k) + v_{3,7}\bar{x}_2(k) + v_{4,7}\bar{x}_1(k) \\ &+ v_{5,7}\bar{x}_{10}(k) + v_{6,7}\bar{x}_9(k) + 2v_{7,7}\bar{x}_8(k)]\end{aligned} \tag{200}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,9} &= -w_6 \cdot p \cdot [v_{0,7} + v_{1,7}\bar{x}_5(k) + v_{2,7}\bar{x}_4(k) + v_{3,7}\bar{x}_3(k) + v_{4,7}\bar{x}_2(k) \\ &+ v_{5,7}\bar{x}_{11}(k) + v_{6,7}\bar{x}_{10}(k) + 2v_{7,7}\bar{x}_9(k)] - w_7 \cdot p \cdot [v_{0,6} + v_{1,6}\bar{x}_4(k) + v_{2,6}\bar{x}_3(k) \\ &+ v_{3,6}\bar{x}_2(k) + v_{4,6}\bar{x}_1(k) + v_{5,6}\bar{x}_{10}(k) + 2v_{6,6}\bar{x}_9(k) + v_{6,7}\bar{x}_8(k)]\end{aligned} \tag{201}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,10} &= -w_5 \cdot p \cdot [v_{0,7} + v_{1,7}\bar{x}_6(k) + v_{2,7}\bar{x}_5(k) + v_{3,7}\bar{x}_4(k) + v_{4,7}\bar{x}_3(k) \\ &+ v_{5,7}\bar{x}_{12}(k) + v_{6,7}\bar{x}_{11}(k) + 2v_{7,7}\bar{x}_{10}(k)] - w_6 \cdot p \cdot [v_{0,6} + v_{1,6}\bar{x}_5(k) + v_{2,6}\bar{x}_4(k) \\ &+ v_{3,6}\bar{x}_3(k) + v_{4,6}\bar{x}_2(k) + v_{5,6}\bar{x}_{11}(k) + 2v_{6,6}\bar{x}_{10}(k) + v_{6,7}\bar{x}_9(k)] \\ &- w_7 \cdot p \cdot [v_{0,5} + v_{1,5}\bar{x}_4(k) + v_{2,5}\bar{x}_3(k) + v_{3,5}\bar{x}_2(k) + v_{4,5}\bar{x}_1(k) + 2v_{5,5}\bar{x}_{10}(k) \\ &+ v_{5,6}\bar{x}_9(k) + v_{5,7}\bar{x}_8(k)]\end{aligned} \tag{202}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,11} &= w_7 - w_5 \cdot p \cdot [v_{0,6} + v_{1,6}\bar{x}_6(k) + v_{2,6}\bar{x}_5(k) + v_{3,6}\bar{x}_4(k) + v_{4,6}\bar{x}_3(k) \\ &+ v_{5,6}\bar{x}_{12}(k) + 2v_{6,6}\bar{x}_{11}(k) + v_{6,7}\bar{x}_{10}(k)] - w_6 \cdot p \cdot [v_{0,5} + v_{1,5}\bar{x}_5(k) + v_{2,5}\bar{x}_4(k) \\ &+ v_{3,5}\bar{x}_3(k) + v_{4,5}\bar{x}_2(k) + 2v_{5,5}\bar{x}_{11}(k) + v_{5,6}\bar{x}_{10}(k) + v_{5,7}\bar{x}_9(k)]\end{aligned} \tag{203}$$

$$\begin{aligned}\bar{\alpha}_{n_y+n_u,12} &= w_6 - w_5 \cdot p \cdot [v_{0,5} + v_{1,5}\bar{x}_6(k) + v_{2,5}\bar{x}_5(k) + v_{3,5}\bar{x}_4(k) + v_{4,5}\bar{x}_3(k) \\ &+ 2v_{5,5}\bar{x}_{12}(k) + v_{5,6}\bar{x}_{11}(k) + v_{5,7}\bar{x}_{10}(k)]\end{aligned} \tag{204}$$

$$\bar{\alpha}_{n_y+n_u,13} = w_5. \tag{205}$$

13.3 Example: Decomposition of Dynamic Quadratic Neural Unit Models for DDHS

QNU Model, $n_y = 4$ *and* $n_u = 3$

$$\begin{aligned}\hat{a}_1 &= w_{0,1} + w_{1,1}\tilde{y}(k-1) + w_{1,2}\tilde{y}(k-2) + w_{1,3}\tilde{y}(k-3) + w_{1,4}\tilde{y}(k-4) + \\ &w_{1,5}u(k-1) + w_{1,6}u(k-2) + w_{1,7}u(k-3)\end{aligned} \tag{206}$$

$$\begin{aligned}\hat{a}_2 &= w_{0,2} + w_{2,2}\tilde{y}(k-2) + w_{2,3}\tilde{y}(k-3) + w_{2,4}\tilde{y}(k-4) + w_{2,5}u(k-1) \\ &+ w_{2,6}u(k-2) + w_{2,7}u(k-3)\end{aligned} \tag{207}$$

$$\begin{aligned}\hat{a}_3 &= w_{0,3} + w_{3,3}\tilde{y}(k-3) + w_{3,4}\tilde{y}(k-4) + w_{3,5}u(k-1) + w_{3,6}u(k-2) \\ &+ w_{3,7}u(k-3)\end{aligned} \tag{208}$$

$$\begin{aligned}\hat{a}_4 &= w_{0,4} + w_{4,4}\tilde{y}(k-4) + w_{4,5}u(k-1) + w_{4,6}u(k-2) \\ &+ w_{4,7}u(k-3).\end{aligned} \tag{209}$$

Further, for coefficients with respect to input terms of the principle input vector **x**

$$\hat{b}_1 = w_{0,5} + w_{5,5}u(k-1) + w_{5,6}u(k-2) + w_{5,7}u(k-3) \tag{210}$$

$$\hat{b}_2 = w_{0,6} + w_{6,6}u(k-2) + w_{6,7}u(k-3) \tag{211}$$

$$\hat{b}_3 = w_{0,7} + w_{7,7}u(k-3). \tag{212}$$

13.4 Example: Decomposition of Dynamic Linear Neural Unit Models as Nonlinear Control Loop for DDHS

LNU with QNU Feedback controller, $n_y = 4$ *and* $n_u = 3$

$$\hat{a}_1 = w_1 \tag{213}$$

$$\begin{aligned}\hat{a}_2 &= w_2 - w_5 \cdot p \cdot [v_{0,1} + v_{1,1}\tilde{y}(k-2) + v_{1,2}\tilde{y}(k-3) + v_{1,3}\tilde{y}(k-4) + \\ &v_{1,4}\tilde{y}(k-5) + v_{1,5}d(k-2) + v_{1,6}d(k-3) + v_{1,7}d(k-4)]\end{aligned} \tag{214}$$

$$\begin{aligned}\hat{a}_3 &= w_3 - w_5 \cdot p \cdot [v_{0,2} + v_{2,2}\tilde{y}(k-3) + v_{2,3}\tilde{y}(k-4) + v_{2,4}\tilde{y}(k-5) + v_{2,5}d(k-2) \\ &+v_{2,6}d(k-3) + v_{2,7}d(k-4)] - w_6 \cdot p \cdot [v_{0,1} + v_{1,1}\tilde{y}(k-3) + v_{1,2}\tilde{y}(k-4) + v_{1,3}\tilde{y}(k-5) \\ &+v_{1,4}\tilde{y}(k-6) + v_{1,5}d(k-3) + v_{1,6}d(k-4) + v_{1,7}d(k-5)]\end{aligned} \tag{215}$$

$$\begin{aligned}\hat{a}_4 &= w_4 - w_5 \cdot p \cdot [v_{0,3} + v_{3,3}\tilde{y}(k-4) + v_{3,4}\tilde{y}(k-5) + v_{3,5}d(k-2) + v_{3,6}d(k-3) \\ &+v_{3,7}d(k-4)] - w_6 \cdot p \cdot [v_{0,2} + v_{2,2}\tilde{y}(k-4) + v_{2,3}\tilde{y}(k-5) + v_{2,4}\tilde{y}(k-6) + v_{2,5}d(k-3) \\ &+v_{2,6}d(k-4) + v_{2,7}d(k-5)] - w_7 \cdot p \cdot [v_{0,1} + v_{1,1}\tilde{y}(k-4) + v_{1,2}\tilde{y}(k-5) + v_{1,3}\tilde{y}(k-6) + \\ &v_{1,4}\tilde{y}(k-7) + v_{1,5}d(k-4) + v_{1,6}d(k-5) + v_{1,7}d(k-6)]\end{aligned} \tag{216}$$

$$\begin{aligned}\hat{a}_5 &= -w_5 \cdot p \cdot [v_{0,4} + v_{4,4}\tilde{y}(k-5) + v_{4,5}d(k-2) + v_{4,6}d(k-3) \\ &+v_{4,7}d(k-4) - w_6 \cdot p \cdot [v_{0,3} + v_{3,3}\tilde{y}(k-5) + v_{3,4}\tilde{y}(k-6) + v_{3,5}d(k-3) + \\ &v_{3,6}d(k-4) + v_{3,7}d(k-5) - w_7 \cdot p \cdot [v_{0,2} + v_{2,2}\tilde{y}(k-5) + v_{2,3}\tilde{y}(k-6) + \\ &v_{2,4}\tilde{y}(k-7) + v_{2,5}d(k-4) + v_{2,6}d(k-5) + v_{2,7}d(k-6)]\end{aligned} \tag{217}$$

$$\begin{aligned}\hat{a}_6 &= -w_6 \cdot p \cdot [v_{0,4} + v_{4,4}\tilde{y}(k-6) + v_{4,5}d(k-3) + v_{4,6}d(k-4) \\ &+v_{4,7}d(k-5)] - w_7 \cdot p \cdot [v_{0,3} + v_{3,3}\tilde{y}(k-6) + v_{3,4}\tilde{y}(k-7) + v_{3,5}d(k-4) \\ &+v_{3,6}d(k-5) + v_{3,7}d(k-6)]\end{aligned} \tag{218}$$

$$\hat{a}_7 = -w_7 \cdot p \cdot [v_{0,4} + v_{4,4}\tilde{y}(k-7) + v_{4,5}d(k-4) + v_{4,6}d(k-5) + v_{4,7}d(k-6)]. \tag{219}$$

Further, for coefficients with respect to input terms of the principle input vector **x**

$$\hat{b}_1 = w_5 \tag{220}$$

$$\hat{b}_2 = w_6 - w_5 \cdot p \cdot [v_{0,5} + v_{5,5}d(k-2) + v_{5,6}d(k-3) + v_{5,7}d(k-4)] \tag{221}$$

$$\begin{aligned}\hat{b}_3 &= w_7 - w_5 \cdot p \cdot [v_{0,6} + v_{6,6}d(k-3) + v_{6,7}d(k-4)] - w_6 \cdot p \cdot [v_{0,5} + v_{5,5}d(k-3) \\ &+v_{5,6}d(k-4) + v_{5,7}d(k-5)]\end{aligned} \tag{222}$$

$$\hat{b}_4 = -w_5 \cdot p \cdot [v_{0,7} + v_{7,7} d(k-4)] - w_6 \cdot p \cdot [v_{0,6} + v_{5,6} d(k-4) + v_{6,7} d(k-5)] \\ -w_7 \cdot p \cdot [v_{0,5} + v_{5,5} d(k-4) + v_{5,6} d(k-5) + v_{5,6} d(k-6)] \tag{223}$$

$$\hat{b}_5 = -w_6 \cdot p \cdot [v_{0,7} + v_{7,7}] - w_7 \cdot p \cdot [v_{0,6} + v_{6,6} d(k-5) + v_{6,7} d(k-6)] \tag{224}$$

$$\hat{b}_6 = -w_7 \cdot p \cdot [v_{0,7} + v_{7,7} d(k-6)]. \tag{225}$$