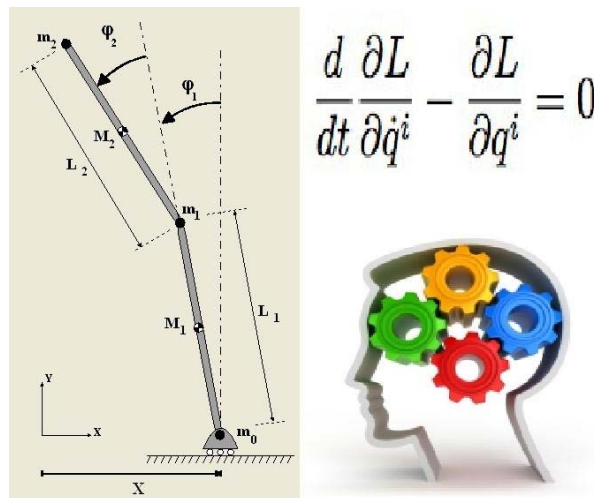


National technical university of Athens



DPMS AUTOMATION CONTROL SYSTEMS

DIPLOMA THESIS



Identification of nonlinear MIMO Lagrange systems using neural networks with guaranteed persistency of excitation

Professor: K. J. Kyriakopoulos

Anagnostis Samanis AM: 021-132-15

Athens, February 2016

ACKNOWLEDGMENTS

This thesis was supported by the Control Systems Laboratory of Mechanical Engineering Department of National Technical University of Athens. I would like to thank the members of the Laboratory for the guidance, help and time they provided me with and for the opportunity that was given to me to get into touch with the interesting fields of systems identification and adaptive control.

CONTENTS

ACKNOWLEDGMENTS	2
LIST OF FIGURES	4
LIST OF PICTURES.....	7
LIST OF TABLES	8
LIST OF ABBREVIATIONS.....	9
ABSTRACT	10
CHAPTER 1 – INTRODUCTION.....	11
Motivation	11
Literature Survey	12
Contributions.....	20
Current work structure.....	21
CHAPTER 2 – PROBLEM STATEMENT & PRELIMINARIES.....	22
A. Lagrangian Dynamics & Problem formulation	22
B. Neural Networks.....	39
C. RBF Neural Networks	40
D. Persistency of Excitation	42
E. Prescribed Performance.....	43
F. Dynamical Systems	45
CHAPTER 3 – MAIN RESULTS	46
A. Reference design	46
B. Control design	55
C. Identification scheme design	65
CHAPTER 4 – SIMULATION RESULTS	83
CHAPTER 5 – CONCLUSIONS & FUTURE WORK.....	121
REFERENCES	123

LIST OF FIGURES

Figure 1. Graphical illustration of the prescribed performance definition.....	page 44
Figure 2. Compact set with the grid of the selected nodes.....	page 47
Figure 3. Optimal ellipses passing from a given point inside a compact set	page 48
Figure 4. Plot of ellipses calculated from the optimization algorithm and for each node there is one ellipse passing through it inside the compact set.....	page 53
Figure 5. 3D Plot of compact set with nodes, ellipses passing through them and RBF	page 54
Figure 6. Phase plane trajectory of system state $x(t)$ in the simulation with 16 nodes.....	page 86
Figure 7. Actual and desired values of x_1 -position of the system as a function of time	page 87
Figure 8. Actual and desired values of x_2 -velocity of the system as a function of time	page 87
Figure 9. Control action guarantees that system states follow the desired trajectory.....	page 88
Figure 10. Plot of W_f estimations calculated by the identifier (16 nodes simulation).....	page 89
Figure 11. Plot of the function estimated by the identification scheme (selecting 16 nodes) along with the function $f(x)$ intended to be estimated	page 90
Figure 12. Plot of errors between the function estimated by the identification scheme (selecting 16 nodes) and the function $f(x)$ intended to be estimated.....	page 91
Figure 13. Plot of function to be estimated by the identification scheme.....	page 92
Figure 14. Plot of estimation of function $f(x)$ calculated from the identification scheme selecting 16 nodes inside the compact set.....	page 92
Figure 15. Phase plane trajectory of system state $x(t)$ in the simulation with 36 nodes.....	page 93
Figure 16. Plot of W_f estimations calculated by the identifier (36 nodes simulation).....	page 95
Figure 17. Plot of estimation of function $f(x)$ calculated from the identification scheme selecting 36 nodes inside the compact set	page 96

Figure 18. Plot of errors between the function estimated by the identification scheme (selecting 36 nodes) and the function $f(x)$ intended to be estimated.....	page 97
Figure 19. Plot of estimation of function $f(x)$ calculated from the identification scheme selecting 36 nodes inside the compact set.....	page 98
Figure 20. Plot of W_f estimations calculated by the identifier (64 nodes simulation)	page 99
Figure 21. Plot of the function estimated by the identification scheme (selecting 64 nodes) along with the function $f(x)$ intended to be estimated	page 100
Figure 22. Plot of errors between the function estimated by the identification scheme (selecting 64 nodes) and the function $f(x)$ intended to be estimated	page 101
Figure 23. Plot of estimation of function $f(x)$ calculated from the identification scheme selecting 64 nodes inside the compact set	page 102
Figure 24. Actual and desired values of x1-position of the system as a function of time ...	page 106
Figure 25. Actual and desired values of x2-velocity of the system as a function of time ...	page 107
Figure 26. Control action guaranteeing that system states follow the desired trajectory ...	page 108
Figure 27. Plot of W_f estimations calculated by the identifier selecting 9 nodes.....	page 109
Figure 28. Plot of W_g estimations calculated by the identifier selecting 9 nodes	page 109
Figure 29. Plot of the function estimated by the identification scheme (selecting 9 nodes) along with the function $f(x)$ intended to be estimated	page 110
Figure 30. Plot of the function estimated by the identification scheme (selecting 9 nodes) along with the function $g(x)$ intended to be estimated	page 111
Figure 31. Plot of the function estimated by the identification scheme (selecting 9 nodes) along with the function $g(x)$ intended to be estimated	page 112
Figure 32. Plot of errors between the function estimated by the identification scheme (selecting 9 nodes) and the function $g(x)$ intended to be estimated	page 113
Figure 33. Plot of estimation of function $f(x)$ calculated from the identification scheme selecting 9 nodes inside the compact set	page 114

Figure 34. Plot of estimation of function $g(x)$ calculated from the identification scheme selecting 9 nodes inside the compact set	page 114
Figure 35. Plot of function $f(x)$ to be estimated by the identification scheme	page 115
Figure 36. Plot of function $g(x)$ to be estimated by the identification scheme	page 115
Figure 37. Plot of W_f estimations calculated by the identifier selecting 16 nodes.....	page 116
Figure 38. Plot of W_g estimations calculated by the identifier selecting 16 nodes	page 117
Figure 39. Plot of the function estimated by the identification scheme (selecting 16 nodes) along with the function $f(x)$ intended to be estimated	page 118
Figure 40. Plot of the function estimated by the identification scheme (selecting 16 nodes) along with the function $g(x)$ intended to be estimated	page 118
Figure 41. Plot of errors between the function estimated by the identification scheme (selecting 16 nodes) and the function $f(x)$ intended to be estimate	page 119
Figure 42. Plot of errors between the function estimated by the identification scheme (selecting 16 nodes) and the function $g(x)$ intended to be estimated	page 120

LIST OF PICTURES

Picture 1. Pendulum (1 DOF Manipulator)	page 32
Picture 2. 2 DOF Robot Manipulator	page 34
Picture 3. Nonlinear model of a neuron	page 39
Picture 4. Block diagram of identification scheme	page 65
Picture 5. Illustration of the key idea behind the proof of the persistency of excitation of $Z_f(x(t))$ and $Z_g(x(t))$	page 66

LIST OF TABLES

Table 1. Results showing the efficiency of learning after three simulations of 16,36,64 nodes	page 102
---	----------

LIST OF ABBREVIATIONS

DNN	Dynamic Neural Network
DOF	Degrees of freedom
FMM	Finite memory machine
FNN	Feedforward Neural Network
ISS	Implicit function theorem
LTV	Linear time varying systems
MIMO	Multiple inputs multiple outputs
MLP	Multilayer perception
MNNs	Multilayer neural networks
NN	Neural Network
PE	Persistency of Excitation condition
PRPE	Parallel Recursive Prediction Error
RBF	Radial Basis Function
RBFs	Radial Basis Functions
RNN	Recurrent Neural Network
SISO	Single input single output
UUB	Uniformly ultimate bounded

ABSTRACT

Motivated by learning methods attempted in previous years and by the recent works in the control literature being done to tackle the problem of identifying nonlinear parts of unknown system dynamics, an effort is presented in this thesis to identify nonlinearities of systems, following Lagrange dynamics, in a local set of values. For this purpose, an exciting technique in a form of autonomous closed loop system is created ensuring a priori that all possible unknown dynamics of the nonlinear function of Lagrange system (to be identified) will be stimulated in a local region of values in order to be successfully learned. This autonomous system produces a desired trajectory, which is designed (inside the local set of identification) in a way to pass through all selected nodes where the centers of Radial Basis Functions (RBF) are put. Additionally the identification architecture is comprised of an on-line Radial Basis Function (RBF) neural network identifier and an adaptive controller. The identifier estimates the unknown dynamics of the system through a process in which, the weight estimator vector of the neural network converges to its optimal values and the learning result is stored in a linear mathematic expression. The adaptive controller is designed in a way to make sure that the system with unknown dynamics follows the desired trajectory given as input to the system, guaranteeing the satisfaction of the persistency of excitation condition for the RBF regressors employed. It is proven in this work that if the persistency of excitation condition is satisfied, then the neural network weights estimates converge to small neighborhoods of their true values, succeeding in learning the actual system nonlinearities effectively. There are two identical characteristics of this approach. The first one is the fact that the autonomous system proposed produces a desired trajectory which is able to stimulate all possible state values of the system to be identified in a local region of values, and the other one is the isolation between the identifier and the controller design increasing the robustness level of the learning scheme. Finally simulation examples of certain systems are provided whose non-linear parts are identified through the identification process explained above and the effectiveness of this method is evaluated through the figures shown in each example.

CHAPTER 1 – INTRODUCTION

Motivation

The concept of learning concerns for over half a century the scientific community and it is widely appeared in the control literature, in which many related papers focus on finding learning methods motivated by needs to estimate or approximate parameters or unknown quantities of nonlinear functions, which are parts of unknown system dynamics. Furthermore the development of learning techniques is arisen from the need to identify unknown controller parameters, in an attempt to achieve regulation under certain control goals and improve system performance. Also learning about environment or learning new design goals and constraints were other usual goals of learning techniques. In primary steps of research made upon the problem of learning, there was an attempt to invent learning techniques in a kind that, the system with unknown dynamics is excited in an open loop form by applying various (mostly arbitrary) inputs. Then by analyzing system outputs and using certain methods (neural network methods usually were applied), an estimation of the unknown parts of the system was tried to be made. Although these efforts achieved some significant results in the literature, they are characterized by the weakness to excite the system in a way to stimulate all possible system dynamics inside the region of interest, where the learning is planned to be made. Consequently the learning is not achieved in a satisfactory way, including all possible system dynamics and it is only achieved in a range of values depending on the applying excitement. The adaptive control techniques used later, boosted the efficiency of learning methods and improved the performance by decreasing the errors, while system followed a desired trajectory or by adjusting control parameters, when operating conditions were changed. Despite the fact that the introduction of adaptive techniques enhanced the learning methods, not any novel exciting method was applied till very lately as a mechanism to stimulate all possible system dynamics meaning a range of all possible positions and velocities of the system in a certain area of interest. There are some recent references (analyzed in the next subsections) in the literature, which tried to produce a kind of orbits, in an attempt to excite as many as possible of the system dynamics in a predefined certain area, without ensuring that all possible system dynamics of this area can be excited a priori, but only a certain percentage of them, resulting in quite a satisfactory result of learning.

In spite of significant learning results have been achieved in the above mentioned efforts, the problem of successful learning of all possible unknown system dynamics in a certain predefined area is still open in the system identification literature. Further steps are needed to be made in this field of research, posing challenges to control engineering community. The successful excitement of all possible dynamics of a nonlinear system in a local predefined area, leads to the complete learning of nonlinear dynamics of the system in this area and this acquired knowledge is expressed by a mathematical model. By the success of this technique, it is feasible to achieve

state estimation of any complicated system and create efficient observers capable of estimating unmeasurable states. The advantage of this achievement can be widely exploited in mechanical systems, which follow Lagrangian dynamics and they are met in everyday life applications. For example, this learning technique mentioned above can be applied in a complex Lagrangian mechanical system like a submarine in a controlled space where suitable equipment can ensure basic state measurements. By acquiring a complete mathematical model from the learning technique (mentioned above), powerful state estimators can be produced for the submarine, giving good estimations in unknown environments under various disturbances like in oceans. Furthermore, by acquiring a good learning result of unknown nonlinear dynamics of a mechanical system, the process of tuning the controller parameters could be facilitated resulting in the achievement of satisfactory control performance. Also in complex systems for which a good mathematical model can be derived, model predictive controllers can be applied succeeding in regulating the system under demanding control goals.

Literature Survey

The approaches which have been used over the years, dealing with the problem of learning are separated into two categories: deterministic and statistical/stochastic learning. Despite the fact that statistical learning could provide satisfactory algorithms for many problems, it cannot be easily used for practical control systems which are deterministic in nature. Also adaptive control techniques, which became popular since 1960, have been introduced in the approaches of learning, contributing to the efficiency of the learning algorithm. Adaptive control method can handle online a control problem despite plant's uncertainties. When the dynamical characteristics of a plant vary considerably, controller parameters are adjusted to achieve desired performance. One basic drawback of adaptive control algorithm is the fact that it does not restore memory, meaning that for any changing situation controller parameters need to be recalculated, so as the system adapts to the new situation, even if the system experienced this same situation before. This means that adaptive control does not have the ability of learning. For this purpose, learning control techniques were developed, having the ability of knowledge acquisition, storage and utilization of stored knowledge for similar future situations. Their main working principle is to design a controller, with the aim to estimate unknown parameters or functions of a plant. If the estimated information converges to the true values, the performance of the controller will be as effective as in the case where the entire plant's information is known. The improvement of the performance depends on the improvement of the estimated unknown information of the system. The learned information is an experience, which is stored and it can be exploited for good performance, when similar situations occur again. Also the process of learning systems can be operated by optimizing a large set of adjustable parameters (or variable structural elements), in order to construct a mapping, which deals with the spatial dependencies of the problem

throughout the operating envelope of the system. In order to execute this optimization learning, systems make extensive use of past information. The basic disadvantage of the learning control techniques is the fact that they require only the adaptive ability, in order to cope with system uncertainties. For this reason it is more than clear that, the processes adaptation and learning are highly-connected and complementary. Especially, adaptive capabilities accommodate slowly time-varying dynamics and novel situations (situation not met before), but they are ineffective for problems with strong nonlinear dynamics. On the other hand, learning control methods are powerful in dealing with poorly modeled nonlinear dynamics, but they are not having good performance in systems with time-varying dynamics. From the above information, it is more than clear that a learning system must be able to accumulate experience and information through closed loop interaction between the plant and the environment, store the information and adapting its stored knowledge in order to operate successfully while meeting new experiences. There is one certain technique which was applied in previous attempts in order to create a learning control system able to exploit previously mentioned capabilities. Especially, a separation of input space into a number of stand-alone without common point regions was made and the specific output of the system is associated with the specific each time input region. This approach, which is not working as a closed loop interaction between the system and the close environment, has a basic drawback. It is the combinatorial growth in the number of regions required, as the state space dimension or number of partitions inside state space is increased.

The effort of control engineers to devise algorithms with adaptation and learning capabilities, led to the development of intelligent control. Intelligent control is a general and more advanced method of learning control, which investigates systems with learning abilities which are able to successfully interact with the environment. One related field of the intelligent control, which is highly connected with the concept of learning, is the neural networks (NN). Neural networks (NN) have proven to be an extremely important tool, achieving effective results of identification, estimation and control of strong and elaborate nonlinear systems with uncertain parts. Significant amounts of researching works in the last few decades are focused on efforts to develop system identification techniques using neural networks.

In [4] neural networks are used for trajectory control of robot manipulator. There is a model comprised of a feedback loop, an internal neural model of motor system and an internal neural model of inverse dynamics of the motor system. In order to generate a motor command to achieve desired trajectory, inverse dynamics must be calculated through inversion of a matrix, which represents the dynamics of the robot manipulator. Since the exact dynamics of the control objects (arms, limbs) are unknown, then the dynamics matrix is also unknown and consequently it is practically inevitable to calculate the inverse dynamics matrix. So a neural network model is proposed and a feedback error learning rule, by which an internal neural model of inverse dynamics of the motor system is acquired during execution of movement. Although neither strict modeling of the manipulator nor precise parameter estimation was required, the control performance by the neural network model was improved gradually during 30mins of learning.

Ronald M. Holdaway and Mark W. White [5] proposed a neural network processing scheme which utilizes a self-organizing map as the front end to a feedforward classifier network. The results of a series of benchmarking studies based upon artificial statistical pattern recognition tasks indicate that the proposed architecture performs significantly better than conventional feedforward classifier networks when the decision regions are disjoint. In [6] multilayer networks are considered with one hidden layer of Gaussian units and an output layer of conventional units. It is shown that for this kind of networks, a fast dimensionality analysis is possible to be performed, by analyzing only a small fraction of the input patterns. As a result of this approach, it is possible to initialize the weights of the network before starting the back propagation training. J. Yuh proposes a multilayered neural network controller [7] with error estimation by the critic equation. A parallel recursive prediction error (PRPE) algorithm is used to train the neural network. The presented control scheme does not require any information about the system. Results of case studies with two dynamic systems showed that, the presented control system has the capability of learning and adapting to the system parameter changes without lengthy off-line training period. Charles W. Lee introduces a method of learning using Neural Networks [8] and adjusting weights by moving to the tangent planes to constraint surfaces. This method gives faster convergence than the propagation gradient descent learning algorithm and preserves the system of back propagating errors through the network. In the paper of C. Lee Giles, B.G. Horne and T. Lin [9], it is shown that a type of Neural Network (Narendra & Parthasarathy, 1990, IEEE Trans. Neural Networks, 1, 4-27) which has feedback but no hidden state neurons can learn a special type of FSM called a finite memory machine (FMM) under certain constraints while giving input values randomly chosen from a uniform distribution in a certain predefined range. In [10] two algorithms are presented, using recurrent neural networks (RNN), achieving nonlinear modeling of mechanical systems. These learning algorithms are based on the quasi-Newton methods that estimate the inverse Hessian of an objective function from the gradient to enable Newton-like optimization algorithms.

In most of the works in control literature based on neural network control of nonlinear systems, the unknown nonlinear system dynamics are approximated by linear or nonlinear parameterized neural networks such as radial basis functions (RBF) networks and multilayer neural networks (MNNs) [11] which were mentioned in the works described in the above paragraph. In addition many of the learning approaches in the above paragraph make use of feedforward neural networks and recurrent neural networks. A feedforward neural network (FNN) is a neural network without any feedback connections. Much progress in learning algorithms for such networks have been made, since the error back-propagation was brought to the attention of the community by Rumelhart. Because of their capability of representing very complex nonlinear mappings, FNNs have been successfully applied to domains such as pattern recognition and curve fitting. On the other hand, a neural network with feedback connections is a recurrent neural network (RNN). Attention has recently turned to RNNs, which have important capabilities not found in FNNs. These capabilities include attractor dynamics and the ability to deal with dynamic

systems through their own dynamics. The use of RNNs will be of particular importance to various problems such as the identification and control of nonlinear dynamic systems. Therefore, the development of efficient learning algorithms is desirable. While the identification and control schemes proposed in the above papers manage to achieve learning in a satisfactory level, a basic disadvantage in all the approaches is the fact that the system is excited in many cases randomly and arbitrarily without a closed loop mode. Also in many neural network control and identification schemes, optimization techniques were used to produce parameter adaptation laws. A basic drawback of the optimization technique is the difficulty to insure stability analysis and performance evaluation of the closed-loop system. This problem is certainly surpassed by the use of adaptive neural network control approaches. These techniques include: a) the design and analysis based on the Lyapunov stability theory, b) the fact that stability and performance of the closed-loop control system can be readily determined and c) neural network weights are tuned online, using a Lyapunov synthesis method, rather than optimization techniques. Summarizing all the conclusions mentioned above, it can be clearly stated that intelligent control method includes two basic characteristics, which makes them identical compared with the previously learning approaches in previous decades. Especially, these are the ability of learning through closed-loop stable systems and also being able to exploit the acquired knowledge in same or similar control situations with closed-loop stability and improved performance.

It has been proved in the literature that, Persistency of Excitation (PE) is a condition whose satisfaction guarantees the achievement of parameter convergence and the success of identification learning algorithms using neural networks. Paper [32] provides conditions under which Persistency of Excitation could be satisfied. Especially it is shown that, if the regressor vector in the neural network scheme is constructed out of radial basis function approximants, it will be persistency excited provided a kind of “ergodic” condition is satisfied. A basic problem regarding the Persistency of Excitation is the fact that it cannot be satisfied a priori for identification of nonlinear systems [11]. Many attempts have been made to overcome this problem. Paper [12] presents a closed-loop autonomous method of identification of nonlinear dynamical systems, using a dynamical version of localized RBF neural networks. RBF networks have been widely used in identification and adaptive control of nonlinear systems due to their universal function approximation ability [33]. An RBF network can be presented in a linear form as a product of neural weight vector and regressor vector. The components of the regressor vector are nonlinear functions of inputs to the RBF network. In paper [34] it is shown that, if the inputs to the RBF network coincide with the network neuron centers, then the corresponding regressor vector satisfies the PE condition. As this is very difficult to be achieved for random input trajectories it has been shown in [35], that the corresponding regressor vector is persistently exciting provided that the input variables to the RBF networks belong to certain neighborhoods of the neuron centers. Using the above proofs of papers [34-35] in paper [12], the employed RBF network is constructed out of centers scattered on a regular lattice which is large enough to contain all the possible states of the dynamical system which is going to be identified. Periodic

(or periodic like) inputs are inserted in the RBF networks and it is shown that there exists a class of nonlinear regressor vector (called regressor subvector) which can lead to the satisfaction of the Persistency of Excitation. This condition is called partial Persistency of Excitation. Centers of the RBF's subvector which satisfy the PE condition, are the ones located in a small neighborhood of the input trajectory. So identification of the unknown dynamics can be achieved within a local region along the periodic trajectory.

In papers [13] and [15] a neural network control scheme is presented, by which an adaptive neural controller is able to make the unknown dynamical system follow a periodic orbit and during that time the learning mechanism constructed out of radial basic functions achieves learning of the unknown dynamics in the neighborhood along the trajectory for which a subvector of the regressor vector achieves the persistency of excitation condition. Paper [13] is dealing with a general form of dynamic nonlinear system while paper [15] is dealing with a dynamical system in Brunovsky form with unity control gain. Also same papers [13,15] provide an analysis of how the learned knowledge can be effectively exploited to achieve stability and improved performance. In [16] a deterministic learning method is developed from neural control of general Brunovsky nonlinear systems, in which the affine terms are unknown nonlinear smooth functions of system states. Due to the unknown affine terms, the existing stability results of adaptive control cannot be applied directly to deterministic learning from neural control of the general Brunovsky system. This problem is solved by presenting a result on stability analysis of linear time varying (LTV) systems. Afterwards by taking a state transformation the closed-loop control system is transformed into an LTV form, for which exponential stability can be guaranteed when the PE condition is satisfied and local accurate approximation of the closed-loop control system dynamics can be achieved along a periodic orbit. Another work of learning is presented in paper [14], where deterministic learning is achieved for strict-feedback systems with unity control gain. The closed-loop system is decomposed into interconnected cascade subsystems with the NNs located in different subsystems. The partial PE condition for each subsystem is satisfied in an iterative manner and accurate approximation of closed-loop control system dynamics is achieved. In an extension of this work in paper [17], deterministic learning of nonlinear strict-feedback systems is presented. One basic problem in this attempt is that, affine terms of strict-feedback systems (which are unknown smooth nonlinear functions of system states) are not appropriately dealt with, and may amplify the NN approximation error, which implies that accurate approximation (learning) cannot be implemented. The problems caused by affine terms and low low-triangular strict feedback structure, are solved by employing a decomposition strategy from paper [14] and a state transformation method from paper [16]. The problem is transformed into the stability analysis of a class of LTV subsystems and it is proven that, RBFs employed in different subsystems can satisfy the partial PE condition and all subsystems are exponentially stable. So, locally-accurate approximation (learning) of closed-loop system dynamics is achieved along the periodic orbits of closed-loop signals. An application of the above deterministic learning theory is presented in paper [18], where learning is achieved from Neural Control in

Motor Systems, when accurate tracking is achieved via adaptive neural control and satisfaction of partial persistency of excitation.

Deterministic learning of nonlinear systems in normal form (nth-order Brunovsky systems with unknown control gain) is attempted in paper [19]. Firstly in order to overpass the difficulty arisen by dealing with the unknown affine terms, a transformation is introduced and a model reference adaptive (neural) control system can be considered as a perturbed linear time-varying (LTV) system. Also unknown affine terms raise an issue of stability, which is proven after using an extension of the result on stability analysis of a class of LTV systems in paper [36]. By applying the above techniques exponential stability of the perturb LTV system is achieved, as well as convergence of partial neural weights. Finally, accurate approximation of the closed-loop dynamics is achieved in a local region along a recurrent orbit of closed-loop signals. Paper [20] uses the technique of deterministic learning of nonlinear systems applied in paper [19] and extends the research in the investigation of performance in the learning speed and learning accuracy. It is shown that, the learning speed and accuracy increases with the level of excitation and decreases with the upper bound of PE. By analyzing the effects of the design parameters an optimal learning speed is shown to exist.

The above deterministic learning method is applied for learning control of ocean surface ship using neural networks in papers [21, 24]. A neural learning control is developed for trajectory tracking of ocean ship dynamics. Under the satisfaction of PE condition the adaptive NN control is shown to be able of learning the uncertain ship dynamics and neural learning control using the knowledge obtained from deterministic learning is constructed to achieve closed-loop stability and improved control performance. Same deterministic learning is applied in papers [22, 25] for learning control of uncertain ship dynamics with the difference of using a high-gain observer to estimate the unmeasurable states of ship dynamics.

The deterministic learning literature were developed more and specifically the paper [23] dealt with learning of nonlinear systems in Strict-feedback form. This paper is an extension of previous work in papers [17,19], which uses an ISS-modular adaptive neural tracking control method by combining backstepping with ISS and small gain theorem. The control scheme guarantees the control-loop stability and ensures via the satisfaction of the partial PE, that the system output converges to a small neighborhood around recurrent reference input signals. Also similar methods developed in previous paper are also used in paper [26], which deals with learning from adaptive neural control of a class of nonlinear pure-feedback systems with unknown affine terms. An adaptive neural controller guarantees the stability and tracking performance of the closed-loop system, by combining the implicit function theorem (ISS) the mean value theorem and backstepping. Also a system decomposition is applied, which divides the stable closed-loop system into a series of interconnected closed-loop subsystems. With a recursive design all of the neural network inputs are recurrent orbits along reference orbits satisfying the partial PE condition. Also a state transformation is introduced to overcome the difficulty caused by non-

affine terms, turning the closed loop sub-systems into the LTV subsystems with small perturbations and through their exponential stability, we obtain exponential convergence of all LTV subsystems and learning, which is stored in constant RBF networks.

Paper [27] is an extension of previous work made in paper [23] and focused on learning control of n -th order Strict-feedback systems, using adaptive neural dynamic surface control. In strict-feedback systems of n -th order, the intermediate variables used as neural network inputs grow drastically as order increases. This makes learning process difficult using the techniques developed in paper [23]. Problems are overcome in papers [27,30]. A stable adaptive neural dynamic surface controller is proposed, using the auxiliary first-order filters. This control scheme guarantees closed-loop stability, ensuring that system output converges to a small neighborhood around the recurrent reference signals in finite time. The derivative of filter output is used as the neural network input, reducing the dimension of neural network inputs, achieving learning of high-order strict feedback systems. Then after a system decomposition, the closed-loop system is turned into LTV perturbed subsystems and partial PE condition of RBF neural networks is satisfied. By a state transformation, a convergence of partial neural weights is recursively achieved and finally learning of system unknown dynamics is accomplished.

The work of learning of nonaffine nonlinear systems is extended from paper [19] to the paper [28], which focuses on dynamic learning from adaptive neural network control of a class of Nonaffine nonlinear systems in normal form, in which the system input does not appear linearly. In the control design a filtered tracking error is introduced to simplify the tracking control design and the implicit function theorem and main value theorem are used, to obtain the implicit desired control input in an affine form. The proposed stable adaptive neural network tracking control proposed, combines input-to-state stability with the small gain theorem and partial PE condition is satisfied. With a state-transformation introduced, the control system is turned into an LTV system with small perturbations and through its exponential stability, exponential convergence of the system is guaranteed and furthermore accurate approximation of the implicit desired control input is satisfied as well. The learnt knowledge of the desired control input is stored in memory without needing of re-adapting to the controller parameters. Similar learning control technique is used in the papers [29,30], which deals with the dynamic learning of a class of nonlinear systems with predefined neural control performance. The system used is a SISO nonlinear system with unknown control gain. The difficulty caused by the constraint from the predefined performance, is overcome by a novel transformed function, turning the constrained tracking control problem into an unconstrained stabilization control problem. The adaptive neural network control scheme combines a filter tracking error with Lyapunov stability and guarantees that, all closed-loop signals are bounded and tracking error converges to a predefined neighborhood of zero. The internal system states become recurrent signals of reference orbits and the partial PE condition of RBF neural networks is satisfied. By introducing a state transformation, an accurate neural approximation of the unknown system dynamics is achieved and the learned knowledge is stored in memory.

All the above papers [12-31], which were extensively described, attempted successfully to achieve on-line learning of different unknown nonlinear systems in a closed-loop form using neural networks and satisfying the partial PE condition. While convergence of the tracking error to a neighborhood of zero is achieved, depending on explicit design parameters (control gains) and some unknown bounded terms (modeling error), there is not any systematic method, which computes the required upper bounds, making the selection of the design parameters to satisfy steady state behavior, that lead to a priori PE condition satisfaction, practically impossible. This difficulty is overpassed in the above papers, by using localized RBF neural networks centered in a neighborhood along recurrent trajectories and proven to satisfy the PE condition for a certain regression subvector of the RBF. This is called partial PE condition providing learning in the correspondent vicinity of the desired trajectory and not over a whole predefined compact desired set of system's functionality. Also in [34] it was shown that, if the inputs of RBF network coincide with the network neuron centers (in the predefined desired set), the corresponding regressor vector satisfies the persistency of excitation. In the above papers the input trajectories are random, periodic, quasi-periodic, almost-periodic and sometimes chaotic and they are unable to coincide to the neuron centers. Another point to be mentioned is that, from the system dynamics learnt it is difficult to reproduce the system nonlinear functions. Additionally it is generally very difficult to reconstruct the actual system nonlinearities after learning. Also in the control approach used, the neural network terms are inserted in the control scheme. Consequently, the control scheme and learning scheme are not isolated raising concerns of the guarantee of closed-loop stability. Despite the fact that closed-loop stability is proved to be satisfied in all the above mentioned papers, there is not any explicit methodology based on design parameter conditions, which ensures the stability of the system. Also not any bounds are provided in the parameters related to the controller design and control action.

Another successful attempt of learning nonlinear structures of system in normal form using RBF neural networks is made in paper [37]. In this work a nonlinear system in canonical form is used with unknown nonlinear control gain. Learning of system nonlinearities is achieved on-line by using a neural network identifier constructed out of RBFs in the regressor vector and an adaptive controller developed using the prescribed performance technique. Prescribed performance control was introduced in papers [38-41], providing a methodology of designing control signals capable of guaranteeing prespecified transient and steady state error performance for various classes of nonlinear systems. In the paper [37] the controller is isolated from the identifier increasing the robustness level of the on-line learning scheme. Furthermore, this paper is motivated by paper [32], which shows that if the regressor vector is constructed out of RBF approximants, it is persistently exciting, provided that a kind of condition is satisfied. So in [37] an ergodic desired trajectory is proposed, traversing all the neuron centers of the selected RBFs inside the predefined compact set, ensuring a priori the persistency of excitation condition. Also the persistency of excitation is guaranteed irrespectively of the RBF regressor utilized to construct the neural network identifier. Using the above methods and also the Lyapunov stability analysis, the neural

network weight estimates are proven to be not only uniformly bounded, but also to converge to a small neighborhood of their true values, achieving the target of learning of the actual system nonlinearities. Also, two different neural networks are used for each nonlinear function of the system making the reconstruction of system nonlinearities feasible. Finally, required upper bounds are defined in the control design, in order to satisfy certain state behavior of the system and furthermore the appropriate conditions for the satisfaction of PE.

Contributions

Paper [37] provides a successful method achieving local learning of structural nonlinearities of a system in canonical form. Based on this method, this thesis presents a methodology proposing a desired trajectory able to include all neurons representatives of possible system functionality in the predefined compact set inside which, the learning is aimed to be done. As a result it can be ensured a priori that, local learning can be achieved in every desired functionality compact system set. Motivated by this issue and also by the papers describing the prescribed performance [38-41], in this thesis there is a methodology proposed of designing a reference trajectory able to traverse all the neuron centers of possible system functionality in a way (ergodic condition) that, the satisfaction of persistency of excitation is verified. Also an adaptive controller based in the prescribed performance technique is developed, which guarantees that system states follow accurately the reference signals and also controller parameter design satisfies a priori the persistency of excitation condition. Furthermore, an online neural network identifier constructed out of RBFs in its regressor vector is designed and as system follows the desired trajectory given as a reference, identifier achieves learning of system nonlinearities successfully. RBF centers coincide with the nodes given in a certain lattice, which represents the desired compact set of system functionality, in which the target of learning is aimed to be achieved. According to a certain theorem for a periodic orbit with given period, if in each time interval this periodic orbit visits a neighborhood (closer than the half of the difference between two consecutive neuron centers) of each neuron center of the RBF network for a minimum amount of time, then regressor vectors are persistently exciting. Controller and identifier designing methods are based in Lyapunov stability analysis, ensuring that the closed loop signals are bounded and that the neural network weight estimates are not only bounded, but also they converge to their true values. All these techniques used above achieving the result of learning are extended in a category of second order systems, following Lagrangian dynamics (representing almost all mechanical systems) with n degrees of freedom. Nonlinearities of the system are expressed as nonlinear functions and one of them is the control gain quantity.

Current work structure

Rest thesis presentation is organized in the following chapters. Chapter two describes the characteristics of the problem this work is concerned with, presenting equations about the structure of the system followed by relevant assumptions. Also relevant theoretical background and mathematical expressions, which are used to develop the method of learning control, are extensively analyzed. Whole steps for methodologies of control design and identification scheme, including all relevant mathematical proofs made, are presenting in chapter three. Next, in the fourth chapter, a simulation of the previous results and proofs is conducted, using certain parameters in a process to verify the methodology proposed and proved in previous chapter. Conclusions and proposals for future work are cited in the final part of this thesis.

CHAPTER 2 – PROBLEM STATEMENT & PRELIMINARIES

A. Lagrangian Dynamics & Problem formulation

Lagrangian Dynamics

The trajectory of the particles of the mechanical systems can be calculated using the Lagrangian dynamics. There are many methods which have been developed as the Lagrange equations in one of two forms, the Lagrange equations of the first kind which handles constraints explicitly and the method using Lagrange multipliers of Lagrange equations of second kind which incorporate constraints by judicious choice of generalized coordinates. In all of these methods the mathematical function derived is called Lagrangian and it is a function of generalized coordinates, their time derivatives and contains all the information about the dynamics of the system.

Lagrangian dynamics does not introduce new laws of physics. It is less general than Newtonian mechanics. Newton's laws can include non-conservative forces like friction; however they must include constraint forces explicitly and are best suited to Cartesian coordinates. On the other hand Lagrangian mechanics is ideal for system with conservative forces and for bypassing constraint forces, and some (not all) non-conservative forces in any coordinate system. Generalized coordinates can be chosen by convenience to exploit symmetries in the system or the geometry of the constraints which considerably simplifies describing the dynamics of the system. Conserved quantities and their symmetries in a direct way are also described by Lagrangian mechanics which is restricted to equilibrium problems.

Lagrangian mechanics is widely used to solve mechanical problems in physics and engineering when Newton's formulation of classical mechanics is not convenient. Lagrange's equations are also used in optimization problems of dynamic systems.

The Lagrangian function includes the dynamics of the entire system. The Lagrangian has no single expression for all physical systems. Any function which generates the correct equations of motion from the Euler-Lagrange equations, in agreement with physical laws is considered as Lagrangian. It is also possible to construct general expressions for large classes of applications. Mathematical expressions for Lagrangian dynamics as cited in [43] are given below.

The non-relativistic Lagrangian equation in a generic form for a system of particles can be defined by:

$$L = T - V \quad (2.1)$$

Where T is the total kinetic energy of the system, and V the potential energy of the system. The dimensions of the Lagrangian are the same as energy. The total kinetic energy of the system of N particles with masses $m_1, \dots, m_N$ is the sum of the kinetic energies of the particles, the energy of the system motion:

$$T = \frac{1}{2} \sum_{k=1}^N m_k v_k^2 \quad (2.2), \quad v_k^2 = \mathbf{v}_k \cdot \mathbf{v}_k = \mathbf{r}_k \cdot \mathbf{r}_k$$

Where $\cdot$ is the dot product. The kinetic energy is a function only of the velocities $\mathbf{v}_k$ not the positions $\mathbf{r}_k$ nor time t , so $T = T(\mathbf{v}_1, \dots, \mathbf{v}_n)$. By contrast, the above expression for velocity shows the kinetic energy in generalized coordinates depends on the generalized velocities, coordinates, and time if the position vectors depend explicitly on time due to time-varying constraints so $T = T\left(q, \frac{dq}{dt}, t\right)$. The potential energy V of the system reflects the energy of interaction between the particles, i.e. how much energy any one particle will have due to all the others and other external influences. For conservative forces, it is a function of the position vectors of the particles only, so $V = V(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n)$. For those non-conservative forces which can be derived from an appropriate potential, the velocities will appear also, $V = V(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{v}_1, \mathbf{v}_2, \dots)$. If there is some external field changing with time, or external driving force, the potential will change with time, so most generally $V = V(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{v}_1, \mathbf{v}_2, \dots, t)$. If the potential or kinetic energy, or both, depend explicitly on time due to time-varying constraints or external influences, the Lagrangian $L\left(q, \frac{dq}{dt}, t\right)$ is explicitly time dependent. If neither the potential nor the kinetic energy depends on time, then the Lagrangian $L\left(q, \frac{dq}{dt}\right)$ is explicitly independent of time. In either case, the Lagrangian will always have implicit time-dependence through the generalized coordinates. In addition to the Lagrangian, it is sometimes possible to introduce another function to account for dissipative forces. In relativistic mechanics, L needs more subtle attention, because it is not the difference between kinetic and potential energy, see relativistic Lagrangian mechanics.

In Newtonian mechanics, the equations are given by Newton's laws. The second law $F = ma$ applies to each particle, so for an N particle system in 3d, there are $3N$ second order ordinary differential equations in the positions of the particles to solve for. In Lagrangian mechanics, the Lagrangian function generates the equations of motion. Although the equations of motion include partial derivatives denoted by $\frac{\partial}{\partial}$, they are still ordinary differential equations in the position coordinates of the particles. The total time derivative denoted $\frac{d}{dt}$ may involve implicit differentiation.

There are two kinds, one which accounts for constraint forces by including the constraint equations, and another which excludes constraint forces and only accounts for non-constraint forces in terms of generalized coordinates. By substituting in the Lagrangian and solving the equations, subject to the initial values of the positions and velocities, will give the positions of the particles as functions of time, and one can see how the system evolves.

In order to produce the equation of motion in this thesis the Lagrange's equation of motion of the second kind was used. According to [44,45] Lagrange's equations of the second kind are mathematical results from the calculus of variations, which can also be used in mechanics.

Substituting in the Lagrangian $L\left(q, \frac{dq}{dt}, t\right)$ gives the equations of motion of the system:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) = \frac{\partial L}{\partial q_j} \quad (2.3) \quad j = 1, \dots, n$$

Where j the degrees of freedom of the relevant system.

The number of equations has decreased compared to Newtonian mechanics, from $3N$ to $n = 3N - C$ coupled second order differential equations in the generalized coordinates. These equations do not include constraint forces at all, only non-constraint forces need to be accounted for. The proof of the above Lagrangian equation (2.3) is presented below.

Proof of the Lagrangian equation of motion of the second kind for conservative systems

Consider a system of particles $k = 1, \dots, N_p$. Let $r^{(k)}$ denote the position of the k^{th} particle relative to an inertial reference frame. Let $f^{(k)}$ be the resultant force acting on the mass m_k of the k^{th} particle. Then, from Newton's Second Law:

$$f^{(k)} = m_k \ddot{r}^{(k)} \quad k = 1, \dots, N_p \quad \text{OR} \quad f^{(k)} - m_k \ddot{r}^{(k)} = 0 \quad (2.4)$$

Let $\delta r^{(k)}$ be a 'virtual' displacement that is compatible with all constraints on the motion of the k^{th} particle. Clearly, any motion $r^{(k)}$ that satisfies (2.4) satisfies the following equation for all $\delta r^{(k)}$.

$$\sum_{k=1}^{N_p} \left(f^{(k)} - m_k \ddot{r}^{(k)} \right) \cdot \delta r^{(k)} = 0 \quad (2.5)$$

Equation (2.5) is, for dynamical systems, the analog of the statement that the virtual work of a system in equilibrium vanishes for all compatible virtual displacements of the system. Consider a decomposition of the resultant force $f^{(k)}$ of the form:

$$f^{(k)} = F^{(k)} + R^{(k)}, \quad k = 1, \dots, N_p$$

Where $R^{(k)}$ is the force due to the constraints and $F^{(k)}$ is the resultant of all other forces acting on the k^{th} particle. Then, because the constraint forces $R^{(k)}$ produce no work during motions compatible with the constraints, i.e. $R^{(k)} \cdot \delta r^{(k)} = 0$, equation (2.5) becomes for all compatible $\delta r^{(k)}$:

$$\sum_{k=1}^{N_p} (F^{(k)} - m_k \ddot{r}^{(k)}) \cdot \delta r^{(k)} = 0 \quad (2.6)$$

Equation (2.6) is the form of Newton's Second Law for constrained dynamical systems that we will use in deriving Lagrange's equations. For holonomic dynamical systems with n degrees of freedom all displacements $r^{(k)}$ can be expressed in terms of n independent generalized coordinates $q_1(t), \dots, q_n(t)$ by means of transformation equations of the form:

$$r^{(k)} = r^{(k)}(q_1(t), \dots, q_n(t), t) \quad k = 1, \dots, N_p$$

Consequently, compatible virtual displacements $\delta r^{(k)}$ can be expressed in terms of virtual changes δq_j in generalized coordinates (with time held constant) by

$$\delta r^{(k)} = \sum_{j=1}^n \frac{\partial r^{(k)}}{\partial q_j} \delta q_j$$

Then, the virtual work δW associated with the virtual displacements $\delta r^{(k)}$ can be written as:

$$\begin{aligned} \delta W &= \sum_{k=1}^{N_p} F^{(k)} \cdot \delta r^{(k)} = \sum_{k=1}^{N_p} \sum_{j=1}^n F^{(k)} \cdot \frac{\partial r^{(k)}}{\partial q_j} \delta q_j \quad \text{or} \\ \delta W &= \sum_{j=1}^n Q_j \delta q_j \end{aligned}$$

Where Q_j are the generalized forces corresponding to the generalized coordinates q_j .

$$Q_j \equiv \sum_{k=1}^{N_p} F^{(k)} \cdot \frac{\partial r^{(k)}}{\partial q_j}$$

Expressing the virtual work term in equation (2.6) in terms of generalized forces and displacements we can rewrite (2.6) in the form:

$$\sum_{k=1}^{N_p} m_k \dot{r}^{(k)} \cdot \delta r^{(k)} = \sum_{j=1}^n Q_j \delta q_j \quad \text{or} \quad \sum_{k=1}^{N_p} m_k \dot{r}^{(k)} \cdot \sum_{j=1}^n \frac{\partial r^{(k)}}{\partial q_j} \delta q_j = \sum_{j=1}^n Q_j \delta q_j$$

Or interchanging the order of summation,

$$\sum_{j=1}^n \left[\sum_{k=1}^{N_p} m_k \dot{r}^{(k)} \cdot \frac{\partial r^{(k)}}{\partial q_j} \delta q_j \right] \delta q_j = \sum_{j=1}^n Q_j \delta q_j \quad (2.7)$$

To derive Lagrange equations we want to show that the term in [] can be expressed in terms of derivatives of the kinetic energy.

$$T = \frac{1}{2} \sum_{k=1}^{N_p} m_k \dot{r}^{(k)} \cdot \dot{r}^{(k)} \quad (2.8)$$

To this end, differentiate equation (2.8) with respect to $\dot{q}_j$ and it is derived:

$$\frac{\partial T}{\partial \dot{q}_j} = \sum_{k=1}^{N_p} m_k \dot{r}^{(k)} \cdot \frac{\partial \dot{r}^{(k)}}{\partial \dot{q}_j} \quad (2.9)$$

Perhaps the least intuitive step in the derivation of the Lagrange's equations is noting that the least term in equation (2.9) satisfies the following identity, known as the “cancellation of the dots”.

$$\frac{\partial \dot{r}^{(k)}}{\partial \dot{q}_j} = \frac{\partial r^{(k)}}{\partial q_j} \quad (2.10)$$

We digress briefly to derive this equation. Differentiation of the transformation equations gives:

$$\dot{r}^{(k)} = \sum_{j=1}^n \frac{\partial r^{(k)}}{\partial q_j} \dot{q}_j + \frac{\partial r^{(k)}}{\partial t} \quad (2.11)$$

Which shows that the function $\dot{r}^{(k)}$ can be expressed as a function of the following form:

$$\dot{r}^{(k)} = \dot{r}^{(k)}(q_1(t), \dots, q_n(t), \dot{q}_1(t), \dots, \dot{q}_n(t), t) \quad (2.12)$$

Differentiation of (2.12) with respect to time gives:

$$\frac{d\dot{r}^{(k)}}{dt} = \sum_{j=1}^n \frac{\partial \dot{r}^{(k)}}{\partial q_j} \dot{q}_j + \frac{\partial \dot{r}^{(k)}}{\partial \dot{q}_j} \ddot{q}_j + \frac{\partial \dot{r}^{(k)}}{\partial t} \quad (2.13)$$

Whereas differentiation of (2.11) with respect to time gives:

$$\frac{d\dot{r}^{(k)}}{dt} = \sum_{j=1}^n \frac{\partial \dot{r}^{(k)}}{\partial q_j} \dot{q}_j + \frac{\partial r^{(k)}}{\partial q_j} \ddot{q}_j + \frac{\partial \dot{r}^{(k)}}{\partial t} \quad (2.14)$$

The first and the third terms in equations (2.13) and (2.14) are identical. The remaining terms can be the same for all motions only if equation (2.10) holds. Thus, the “cancellation of dots” identity is proven. Returning to the derivation of Lagrange’s equations and using the “cancellation of dots” identity in equation (2.9) we find:

$$\frac{\partial T}{\partial \dot{q}_j} = \sum_{k=1}^{N_p} m_k \dot{r}^{(k)} \cdot \frac{\partial r^{(k)}}{\partial q_j} \quad (2.15)$$

Differentiation of equation (2.15) with respect to time gives:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) = \sum_{k=1}^{N_p} m_k \ddot{r}^{(k)} \cdot \frac{\partial r^{(k)}}{\partial q_j} + \sum_{k=1}^{N_p} m_k \dot{r}^{(k)} \cdot \frac{\partial \dot{r}^{(k)}}{\partial q_j} \quad (2.16)$$

Where the last term is simply $\frac{\partial T}{\partial q_j}$. Hence,

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} = \sum_{k=1}^{N_p} m_k \ddot{r}^{(k)} \cdot \frac{\partial r^{(k)}}{\partial q_j} \quad (2.17)$$

Substitution of (2.17) into (2.7) gives:

$$\sum_{j=1}^n \left[\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} - Q_j \right] \delta q_j = 0 \quad (2.18)$$

Because the variations δq_j in equation (2.18) are independent, we conclude that:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} - Q_j = 0 \quad (2.19) \quad , \quad j = 1, \dots, n$$

Equations (2.19) are Lagrange equations for systems with holonomic constraints, whether or not the forces are conservative. If all forces are conservative, then $Q_j = -\frac{\partial V}{\partial q_j}$ and (2.19) becomes:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} + \frac{\partial V}{\partial q_j} = 0 \quad (2.20) \quad , \quad j = 1, \dots, n \quad \text{or} \quad \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial (T - V)}{\partial q_j} = 0 \quad , \quad j = 1, \dots, n$$

Or because $\frac{\partial V}{\partial \dot{q}_j} = 0$, then:

$$\frac{d}{dt} \left(\frac{\partial(T-V)}{\partial \dot{q}_j} \right) - \frac{\partial(T-V)}{\partial q_j} = 0 \quad (2.21) \quad , \quad j = 1, \dots, n$$

Finally, introducing the Lagrangian function $L \equiv T - V$ (2.22), equations (2.21) can be written as:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0 \quad (2.23) \quad , \quad i = 1, \dots, n$$

Equations (2.23) are Lagrange's equations for conservative systems with holonomic constraints. Lagrangian dynamics is verified in all mechanical systems from the simple ones (mass spring system, pendulum) as well as complicated like (robotic manipulator with many degrees of freedom). There are two examples given below which show the derivation of equations of motion for two different mechanical systems using the Lagrange method based on the equations proven above.

Proof of the Lagrangian equation of motion of the second kind for non-conservative systems

Assuming a system of point masses of m_μ total system mass, with n degrees of freedom $q_1, \dots, q_n$ which are generalized movement variables for each degree of freedom separately. The equation giving the movements is given in the form:

$$\underline{\ddot{x}}_\mu = \underline{\ddot{x}}_\mu(q_1, \dots, q_n) \quad (2.24)$$

The total force exerted in the system is given from the equation:

$$F_\mu = m_\mu \underline{\ddot{x}}_\mu \quad (2.25)$$

Differentiating the equation (2.24) it is derived:

$$(2.24) \Rightarrow \dot{\underline{x}}_\mu = \sum_i \left(\frac{\partial \underline{x}_\mu}{\partial q_i} \dot{q}_i \right) \Rightarrow \frac{\partial \dot{\underline{x}}_\mu}{\partial \dot{q}_i} = \frac{\partial \underline{x}_\mu}{\partial q_i} \quad (2.26)$$

The kinetic energy is given by the equation:

$$K = \sum_\mu \frac{1}{2} m_\mu \dot{\underline{x}}_\mu^T \dot{\underline{x}}_\mu \quad (2.27)$$

Assuming the generalized force Q_i given by the following equation:

$$Q_i = \sum_\mu F_\mu^T \frac{\partial \underline{x}_\mu}{\partial q_i} \quad (2.28) \quad \text{as it holds that} \quad Q_i \cdot \delta q_i = \sum_\mu F_\mu^T \delta \underline{x}_\mu \quad (2.29)$$

Then we calculate the quantity $\frac{\partial K}{\partial \dot{q}_i}$

$$\frac{\partial K}{\partial \dot{q}_i} = \sum_{\mu} m_{\mu} \dot{x}_{\mu}^T \frac{\partial \dot{x}_{\mu}}{\partial \dot{q}_i} \quad (2.30)$$

Also from the above equation it is derived that:

$$(2.30) \Rightarrow \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_i} \right) = \sum_{\mu} m_{\mu} \ddot{x}_{\mu}^T \frac{\partial \dot{x}_{\mu}}{\partial \dot{q}_i} + \sum_{\mu} m_{\mu} \dot{x}_{\mu}^T \frac{\partial \ddot{x}_{\mu}}{\partial \dot{q}_i} \quad (2.31)$$

Using the equations(2.28), (2.30) the above equation becomes:

$$Q_i = \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} \quad (2.32)$$

Assuming P the potential energy stored in the mechanical system defined above. For the applied forces in the mechanical system defined above it holds that:

$$F_{\mu} = F_{\mu,P} + F_{\mu,T} \quad (2.33) \text{ and } Q_i = Q_{i,P} + \tau_i \quad (2.34)$$

Where $F_{\mu,P}$ conservative force.

It also holds that, $F_{\mu,P} = -\nabla_{x_{\mu}} P = -\frac{\partial P}{\partial x_{\mu}}$ (2.35) and $Q_{i,P} = \sum_{\mu} F_{\mu,P}^T \cdot \frac{\partial x_{\mu}}{\partial q_i} = -\frac{\partial P}{\partial q_i}$ (2.36) and $\frac{\partial P}{\partial \dot{q}_i} = 0$

Substituting (2.34) and (2.35) into (2.31) it is derived that:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_i} \right) &= \sum_{\mu} (F_{\mu,P} + F_{\mu,T})^T \cdot \frac{\partial \dot{x}_{\mu}}{\partial \dot{q}_i} + \frac{\partial K}{\partial q_i} = \sum_{\mu} F_{\mu,P}^T \frac{\partial \dot{x}_{\mu}}{\partial \dot{q}_i} + \tau_i + \frac{\partial K}{\partial q_i} \Rightarrow \\ \tau_i &= \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i} \Rightarrow \tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (2.37) \end{aligned}$$

The above equations are called Lagrange equations or Lagrange model for non-conservative systems.

General form of equations of motion of mechanical systems using Lagrange equations

In this part a generic form of equations of motion of mechanical systems is calculated using the Lagrange equations (2.23) proven in the previous subsections firstly for conservative systems for which the Lagrange model is in the form:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0 \quad (2.38), \quad i = 1, \dots, n$$

Also for non-conservative systems the Lagrange model is in the form:

$$\tau_i = \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i} \Rightarrow \tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (2.39), \quad i = 1, \dots, n$$

Where n the degrees of freedom of the system.

Assuming a system of point masses of m_μ total system mass, with n degrees of freedom $q_1, \dots, q_n$ which are generalized movement variables for each degree of freedom separately. The kinetic energy is calculated from the equation:

$$K_i = \frac{1}{2} m_i v_{ci}^T v_{ci} + \frac{1}{2} \omega_{ci}^T I_i \omega_{ci} \Rightarrow K = \sum_{i=1}^n (K_i) \quad (2.40)$$

It holds that:

$$\left\{ \begin{array}{l} v_{ci} = J_{L_i}^{(i)} \cdot \dot{q}_1 + \dots + J_{L_i}^{(i)} \cdot \dot{q}_i \\ \omega_{ci} = J_{A_i}^{(i)} \cdot \dot{q}_1 + \dots + J_{A_i}^{(i)} \cdot \dot{q}_i \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} v_{ci} = J_L^{(i)} \cdot \dot{q} \\ \omega_{ci} = J_A^{(i)} \cdot \dot{q} \end{array} \right\} \quad (2.41)$$

$$\text{Where } J_L^{(i)} = \left[\underbrace{\underline{J}_{L_1}^{(i)} \quad \dots \quad \underline{J}_{L_i}^{(i)}}_i \quad \underbrace{\underline{0} \quad \dots \quad \underline{0}}_{n-i} \right] \text{ and } J_A^{(i)} = \left[\underbrace{\underline{J}_{A_1}^{(i)} \quad \dots \quad \underline{J}_{A_i}^{(i)}}_i \quad \underbrace{\underline{0} \quad \dots \quad \underline{0}}_{n-i} \right]$$

$$\underline{J}_{L_j}^{(i)} = \left\{ \begin{array}{l} \hat{b}_{j-1} : \text{prismatic} \\ \hat{b}_{j-1} \times r_{j-1, c_i}^{(0)} : \text{rotational} \end{array} \right\}, (j = 1, \dots, i) \text{ and } \underline{J}_{L_j}^{(i)} = \left\{ \begin{array}{l} 0 : \text{prismatic} \\ \hat{b}_{j-1} : \text{rotational} \end{array} \right\}, (j = 1, \dots, i)$$

Consequently the calculation for the kinetic energy can take the form:

$$K = \frac{1}{2} \sum_{i=1}^n \left(m_i \dot{q}^T \cdot J_L^{(i)T} \cdot J_L^{(i)} \cdot \dot{q} + \dot{q}^T \cdot J_A^{(i)T} I_i \cdot J_A^{(i)} \cdot I_i \cdot J_A^{(i)} \cdot \dot{q} \right) = \frac{1}{2} \dot{q}^T D(q) \dot{q}$$

$$\text{Where } D = \sum_{i=1}^n \left(m_i \cdot J_L^{(i)T} \cdot J_L^{(i)} + J_A^{(i)T} I_i \cdot J_A^{(i)} \right) \quad (2.42) \text{ the inertia matrix.}$$

So, the equation (2.39) becomes:

$$\tau = \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}} \right) - \frac{\partial K}{\partial q} + \frac{\partial P}{\partial q} \Rightarrow \tau = \frac{d}{dt} (D(q) \dot{q}) - \frac{\partial K}{\partial q} + \frac{\partial P}{\partial q} \Rightarrow$$

$$\Rightarrow \tau = D(q)\ddot{q} + \dot{D}(q)\dot{q} - \frac{1}{2}\dot{q}^T \frac{\partial D}{\partial q} \dot{q} + \frac{\partial P}{\partial q} \quad (2.43)$$

Where $h(q, \dot{q}) = \dot{D}(q)\dot{q} - \frac{1}{2}\dot{q}^T \frac{\partial D}{\partial q} \dot{q} + \frac{\partial P}{\partial q}$ (2.44) the matrix of centripetal & coriolis torques and

$g(q) = \frac{\partial P}{\partial q}$ the gravitational torques matrix.

For the kinetic energy it can be written:

$$K = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n D_{jk} \cdot \dot{q}_j \cdot \dot{q}_k \quad (2.45)$$

So, for equation (2.43) it holds:

$$\begin{aligned} \tau_i &= \frac{d}{dt} \left[\sum_{j=1}^n (D_{ij}(q) \cdot \dot{q}_j) \right] - \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial D_{jk}}{\partial q_i} \cdot \dot{q}_j \cdot \dot{q}_k \right) + \frac{\partial P}{\partial q_i} \Rightarrow \\ \tau_i &= \sum_{j=1}^n (D_{ij}(q) \cdot \ddot{q}_j) + \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial D_{ij}}{\partial q_k} \cdot \dot{q}_j \cdot \dot{q}_k \right) - \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial D_{jk}}{\partial q_i} \cdot \dot{q}_j \cdot \dot{q}_k \right) + \frac{\partial P}{\partial q_i} \quad (2.46) \end{aligned}$$

Where:

$$D(q)\ddot{q} = \sum_{j=1}^n (D_{ij}(q) \cdot \ddot{q}_j) \quad (2.47)$$

$$h_i(q, \dot{q}) = \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial D_{ij}}{\partial q_k} \cdot \dot{q}_j \cdot \dot{q}_k \right) - \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial D_{jk}}{\partial q_i} \cdot \dot{q}_j \cdot \dot{q}_k \right) \quad (2.48)$$

$$g_i(q) = \frac{\partial P}{\partial q_i} \quad (2.49), \text{ where } P = \sum_{j=1}^n (m_j \hat{g}^T \vec{r}_{0,cj})$$

So it holds that:

$$h(q, \dot{q}) = [h_i(q, \dot{q})]_{i=1, \dots, n} = \left[\sum_{j=1}^n \sum_{k=1}^n (h_{ijk}(q, \dot{q}) \dot{q}_j \dot{q}_k) \right]_{i=1, \dots, n} \quad (2.50)$$

$$\text{Where } h_{ijk}(q, \dot{q}) = \frac{\partial D_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial D_{jk}}{\partial q_i} \quad (2.51)$$

$$\text{Also } g_i(q) = \frac{\partial P}{\partial q_i} = \sum_{j=1}^n m_j \hat{g}^T \frac{\partial \vec{r}_{0,cj}}{\partial q_i} = \sum_{j=1}^n m_j \hat{g}^T J_{Li}^{(j)}(q) \quad (2.52)$$

As a synopsis the **generic form of dynamic Lagrange model** is:

$$\tau = D(q)\ddot{q} + h(q, \dot{q}) + g(q) \quad (2.53)$$

Where $D(q)$ the matrix of inertia as calculated in equation (2.42) and $h(q, \dot{q})$ the matrix of centripetal & coriolis torques, as calculated in the equation(2.50). Alternatively for $h(q, \dot{q})$ it can also be written:

$$h(q, \dot{q}) = \left[\underbrace{C_{ij}(q, \dot{q})}_{C(q, \dot{q})} \right] \dot{q} \quad (2.54)$$

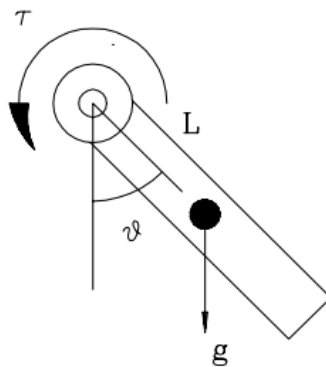
$$\text{Where } C_{ij}(q, \dot{q}) = \sum_{k=1}^n (h_{ijk}(q) \dot{q}_k) = \sum_{k=1}^n \left(\left(\frac{\partial D_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial D_{jk}}{\partial q_i} \right) \dot{q}_k \right) \quad (2.55)$$

Also, $g(q)$ the gravitational torques matrix. This matrix can be calculated as:

$$g(q) = [g_i(q)]_{i=1, \dots, n} \quad (2.56), \text{ where } g_i(q) \text{ as calculated from equation (2.52).}$$

Example 1 – Dynamic model of a pendulum (1DOF Manipulator)

Consider the pendulum mechanical system which is shown in the below picture.



Picture 1. Pendulum (1 DOF Manipulator)

Lagrange dynamics will be used to calculate the equations of motion for the above system. Consider:

θ ... Joint variable

τ ... Joint torque

m ... Mass

L ... Distance between center of mass and joint

d ... Viscous friction coefficient

Firstly the Kinetic energy is calculated:

$$T = \frac{1}{2} I \dot{\theta}^2 \quad (2.57)$$

Secondly the Potential energy is calculated:

$$V = mgL(1 - \cos \theta) \quad (2.58)$$

The Lagrange function (equation(2.22)) is:

$$L = \frac{1}{2} I \dot{\theta}^2 - mgL(1 - \cos \theta) \quad (2.59)$$

Next the following quantities are calculated:

$$\frac{\partial L}{\partial \dot{\theta}} = I \dot{\theta} \quad , \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} = I \ddot{\theta} \quad , \quad \frac{\partial L}{\partial \theta} = -mgL \sin \theta$$

The generalized Lagrangian force in this case must account for the torque applied to the joint and for the friction effect:

$$\psi = \tau - d \dot{\theta}$$

From the general expression:

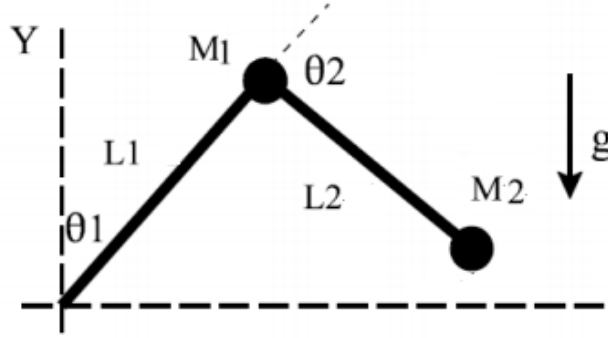
$$\psi = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta}$$

The following second order differential equation is derived:

$$I \ddot{\theta} + d \dot{\theta} + mgL \sin \theta = \tau \quad (2.60)$$

Example 2 – Dynamic model of a two degrees of freedom of Robotic Manipulator

In [46] a robotic manipulator of two degrees of freedom is presented whose equations of motion are given using Lagrange dynamics. The system of two degrees of freedom robotic manipulator to be examined is given in the below picture:



Picture 2. 2 DOF Robot Manipulator

Firstly a geometric analysis is done:

$$x_1 = L_1 \sin \theta_1$$

$$y_1 = L_1 \cos \theta_1$$

$$x_2 = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

$$y_2 = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2)$$

Kinetic energy is calculated with the following equation:

$$T = \frac{1}{2} M_1 \dot{x}_1^2 + \frac{1}{2} M_1 \dot{y}_1^2 + \frac{1}{2} M_2 \dot{x}_2^2 + \frac{1}{2} M_2 \dot{y}_2^2$$

By simplification, it is derived:

$$T = \frac{1}{2} (M_1 + M_2) L_1^2 \dot{\theta}_1^2 + \frac{1}{2} M_2 L_2^2 \dot{\theta}_1^2 + M_2 L_2^2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} M_2 L_2^2 \dot{\theta}_2^2 + M_2 L_1 L_2 \cos \theta_2 (\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1^2)$$

The potential energy is:

$$V = M_1 g L_1 \cos \theta_1 + M_2 g (L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2))$$

So, by Lagrangian Dynamics we form the Lagrangian equation:

$$L = T - V \Rightarrow$$

$$\Rightarrow L = \frac{1}{2}(M_1 + M_2)L_1^2\dot{\theta}_1^2 + \frac{1}{2}M_2L_2^2\dot{\theta}_1^2 + M_2L_2^2\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}M_2L_2^2\dot{\theta}_2^2 + M_2L_1L_2\cos\theta_2(\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1^2) - M_1gL_1\cos\theta_1 - M_2g(L_1\cos\theta_1 + L_2\cos(\theta_1 + \theta_2))$$

So, forming the dynamics equations to be:

$$f_{\theta_{1,2}} = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_{1,2}} \right] - \frac{\partial L}{\partial \theta_{1,2}}$$

So, the dynamics equations after simplifications become:

$$\left((M_1 + M_2)L_1^2 + M_2L_2^2 + 2M_2L_1L_2\cos\theta_2 \right) \ddot{\theta}_1 + \left(M_2L_2^2 + M_2L_1L_2\cos\theta_2 \right) \ddot{\theta}_2 - M_2L_1L_2\sin\theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) - (M_1 + M_2)gL_1\sin\theta_1 - M_2gL_2\sin(\theta_1 + \theta_2) = f_{\theta_1}$$

$$\left(M_2L_2^2 + M_2L_1L_2\cos\theta_2 \right) \ddot{\theta}_1 + M_2L_2^2\ddot{\theta}_2 - M_2L_1L_2\sin\theta_2\dot{\theta}_1\dot{\theta}_2 - M_2gL_2\sin(\theta_1 + \theta_2) = f_{\theta_2}$$

So, we can describe the equations of motion of the system in Matrix form:

$$M(q)\ddot{q} + C(\dot{q}, q) + g(q) = F$$

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

$$M(q) = \begin{bmatrix} \left((M_1 + M_2)L_1^2 + M_2L_2^2 + 2M_2L_1L_2\cos\theta_2 \right) & \left(M_2L_2^2 + M_2L_1L_2\cos\theta_2 \right) \\ M_2L_2^2 + M_2L_1L_2\cos\theta_2 & M_2L_2^2 \end{bmatrix}$$

$$C(\dot{q}, q) = \begin{bmatrix} -M_2L_1L_2\sin\theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ -M_2L_1L_2\sin\theta_2\dot{\theta}_1\dot{\theta}_2 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} -(M_1 + M_2)gL_1\sin\theta_1 - M_2gL_2\sin(\theta_1 + \theta_2) \\ -M_2gL_2\sin(\theta_1 + \theta_2) \end{bmatrix}$$

$$F = \begin{bmatrix} f_{\theta_1} \\ f_{\theta_2} \end{bmatrix}$$

Problem Formulation

The work of this thesis is based on a nonlinear second order Lagrange system of n degrees of freedom. Using the generic form of Lagrange dynamic model equation as proven and defined in equation(2.53), the dynamic model equation examined in this thesis can be written in the following form:

$$M(\underline{x})\ddot{\underline{x}} + C(\underline{x}, \dot{\underline{x}}) + G(\underline{x}) = \underline{u} \quad (2.61)$$

Where:

$$\underline{x} = \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ \vdots \\ x_n \end{bmatrix} \dots, \text{joints displacements}$$

$$\dot{\underline{x}} = \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \vdots \\ \vdots \\ \dot{x}_n \end{bmatrix} \dots, \text{joints velocities}$$

$M(\underline{x}) \dots$, the inertia matrix

$C(\underline{x}, \dot{\underline{x}}) \dots$, the matrix of centripetal & coriolis torques

$G(\underline{x}) \dots$, the gravitational torques matrix

$\underline{u} \dots$, the applied torques inputs (control inputs)

We transform the system in the canonical form.

$$\begin{aligned} \dot{\underline{x}}_1 &= \underline{x}_2 \\ \dot{\underline{x}}_2 &= \underline{f}(\underline{x}) + \underline{g}(\underline{x})\underline{u} \end{aligned} \quad (2.62)$$

Where: $\underline{x} = \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix}$ is the overall state vector and $\underline{x}_1 = [x_1, \dots, x_n]^T$ the vector of positions and

$\underline{x}_2 = [\dot{x}_1, \dots, \dot{x}_n]^T$ the vector of velocities

$f, g: \mathbb{R}^n \rightarrow \mathbb{R}$ are unknown smooth nonlinear functions, f is a matrix with dimensions $n \times 1$, while g is a matrix with dimensions $n \times 2$.

Second equation of (2.62) is transformed to:

$$\underline{g}(x)^{-1} \dot{\underline{x}} = \underline{g}(x)^{-1} \underline{f}(x) + \underline{u} \Rightarrow \underline{g}(x)^{-1} \dot{\underline{x}} - \underline{g}(x)^{-1} \underline{f}(x) = \underline{u} \quad (2.63)$$

From (2.61) and (2.63) it is taken:

$$\underline{g}(x)^{-1} = M(x) \Rightarrow \underline{g}(x) = M(x)^{-1} \quad (2.64)$$

$$\text{and } -\underline{g}(x)^{-1} \underline{f}(x) = C(x, \dot{x}) + G(x) \Rightarrow \underline{f}(x) = -M(x)^{-1} [C(x, \dot{x}) + G(x)] \quad (2.65)$$

The reference model is considered as a function derived by calculating the characteristics of the ellipses which cross neuron centers placed in a lattice in a pre-defined two dimensional compact set of system positions and velocities. The bounds of the desired learning compact set and neuron centers coordinates inside the compact set are given as inputs in the reference model function, and the outputs are two different trajectories of position and velocity which must be followed from system states (position and velocity) of the Lagrangian system. If the Lagrangian system is of more than 1 degree of freedom, then there will be a combination of two different lattices of neuron centers each of one represents the possible state values (position and velocities) for each degree of freedom. Then all ellipses trajectory of each degree of freedom examined lattice is combined with each ellipse from the lattices of rest degrees of freedom, and the combined trajectory arisen is the desired trajectory input which must be followed by the examined degree of freedom states (position and velocity). The same procedure is done for the other degrees of freedoms separately.

$$q_d = F(x_{1u}, x_{1l}, x_{2u}, x_{c1}, \dots, x_{cn}, y_{c1}, \dots, y_{cn}) \quad (2.66)$$

Where:

q_d ..., the desired trajectory arisen from the reference model

x_{1u} ..., the upper bound of the position values of the desired compact set

x_{1l} ..., the lower bound of the position values of the desired compact set

$\pm x_{2u}$..., the upper/lower bounds of the velocity values of the desired compact set

$[(x_{c1}, y_{c1}), \dots, (x_{cn}, y_{cn})]$..., the coordinates of the neuron centers placed in the desired compact set

The objective is to learn nonlinear functions f, g locally in a chosen compact set, by designing a closed loop identification scheme which gives local accurate approximations of the nonlinear functions over the compact set. Following assumptions are exploited in solving this problem.

Assumption 1: There exists an unknown positive constant g^* such that: $|g(x)| \geq g^* > 0, \forall x \in \mathfrak{R}^n$

Assumption 2: The sign of $g(x)$ is considered known, without loss of generality we assume it is positive.

Assumption 3: The inertia matrix $M(q)$ is symmetric, positive definite and satisfies the following inequality $\forall \xi(t) \in \mathfrak{R}^n$:

$$m_1 \|\xi\|^2 \leq \xi^T M(q) \xi \leq \bar{m}(q) \|\xi\|^2 \quad (2.67)$$

Where:

$m_1 \in \mathfrak{R}_{++}$, is a known positive constant

$\bar{m}(q) \in \mathfrak{R}_{++}$, is a known positive function and $\|\cdot\|$ denotes the standard Euclidean norm.

Assumption 4: The following skew-symmetric relationship is satisfied:

$$\xi^T (\dot{M}(q) - 2V_m(q, \dot{q})) \xi = 0 \quad \forall \xi \in \mathfrak{R}^n \quad (2.68)$$

Assumption 5: If $q(t), \dot{q}(t) \in \mathcal{L}_\infty$ then $C(q, \dot{q})$ and $G(q)$ are bounded. Moreover if $q(t), \dot{q}(t) \in \mathcal{L}_\infty$ then the first and second partial derivatives of the elements of $M(q)$, $C(q, \dot{q})$, $G(q)$ with respect to $q(t)$ exist and they are bounded and the first and second partial derivatives of the element of $C(q, \dot{q})$ with respect to $\dot{q}(t)$ exist and they are bounded.

Assumption 6: The desired trajectory is assumed to be designed such that $q_d(t), \ddot{q}_d(t) \in \mathfrak{R}^n$ exist and they are bounded.

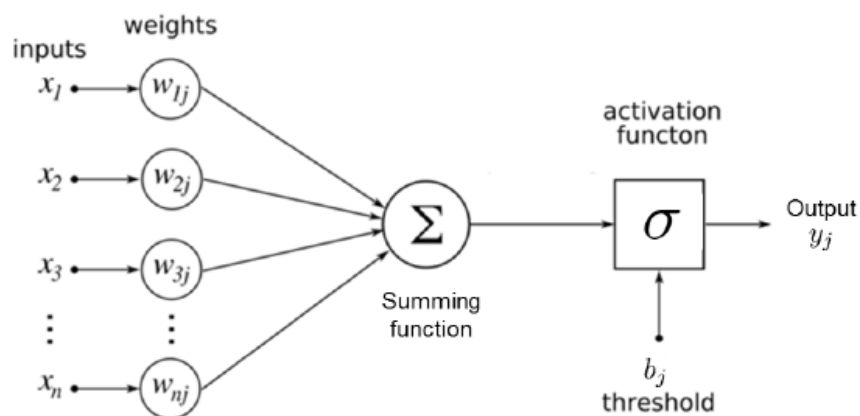
B. Neural Networks

In this subsection taking information from [48] a brief background on artificial NNs is provided. NN structures, learning methods, and approximation properties are included. The structures of both multilayer NNs and DNNs are described. A NN is a massively parallel distributed processor composed of simple processing units that have a natural propensity for storing experiential knowledge and making it available for use. A NN resembles the brain in two respects:

- Knowledge is acquired by the network from its environment through a learning process.
- Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

The procedure for the learning process is called a learning algorithm. Its function is to modify the synaptic weights of the network to attain a desired design objective. The weight modification equips the method for NN design and implementation. Facilitating identification, estimation, and control for a wide class of nonlinear systems, NNs offer several useful properties and capabilities:

1. **NONLINEARITY.** A NN is constructed from an interconnection of nonlinear neurons, so it is itself nonlinear. This is an important property, especially if the underlying mechanism is inherently nonlinear.
2. **INPUT-OUTPUT MAPPING.** The synaptic weights (free parameters) of the network are modified to minimize the difference between the desired response and the actual response of the network produced by the input signal in accordance with appropriate criterion. Hence, the NN can adapt to construct the desired input - output mapping.
3. **ADAPTIVITY.** NNs have a built-in capability to adapt their weights based on design criterion. The on-line learning algorithm can lead the network to adapt in real time.



Picture 3. *Nonlinear model of a neuron*

A neuron is the fundamental unit for the operation of a NN. Its model is shown in Picture 3 with three basic elements:

- a) A set of synapses links, with each element characterized by its own weight.
- b) An adder for summing the input signals multiplied by their respective weights.
- c) A nonlinear activation function transforming the adder output into the output of the neuron.

In mathematical terms, the neuron can be described as:

$$y_j = \sigma \left(\sum_{i=1}^n w_{ij} x_i + b_j \right) \quad (2.69)$$

Where $x_1, x_2, \dots, x_n$ are the input signals, $w_{1j}, w_{2j}, \dots, w_{nj}$ are the respective synaptic weights of neuron j , b_j is the bias term, $\sigma(\cdot)$ is the activation function and y_j is the output signal of the neuron. The activation function $\sigma(\cdot)$ is often chosen as hard limit, linear threshold, sigmoid, hyperbolic tangent, augmented ratio of squares, or radial basis functions

The way in which neurons of a Neural Network (NN) are interconnected determines its structure.

C. RBF Neural Networks

In work [42] it is cited that the multilayer perception (MLP) with sigmoidal activation function and trained via back-propagation algorithm and the radial basis function network (RBF) with Gaussian functions, are the two mostly appeared neural networks form in the artificial neural network literature. Both of them have achieved considerable success in the engineering problem of estimation of nonlinear functions.

Also in the book [3] it is mentioned RBF neural networks were motivated by the locally tuned response observed in biological neurons, and they are extensively used up to now in engineering application of estimation and control of nonlinear systems. They are constructed of two-layer networks in which the hidden layer performs a fixed nonlinear transformation with no adjustable parameters. The output layer combines the outputs in the latter space in a linear form. This linear combination of typically nonlinear basis functions is a key to the RBF representational ability while maintaining computational and analytical tractability. Combining the relevant theoretical information regarding RBFs which is presented in paper [37] and book [3], the basic mathematical background used in this work about RBFs is given below.

RBFs belong to a category of linearly parameterized networks, and this is their representational mathematical form:

$$f(x) = \sum_{i=1}^p w_i z_i(x) = W^T Z(x) \quad (2.70)$$

Where $x \in \mathfrak{R}^n$ is the input vector and $f(x) \in \mathfrak{R}$ is the output. $W = [w_1, \dots, w_n]^T \in \mathfrak{R}^n$ is the weight vector, $n > 1$ is the neural network node number and $Z(x) = [z_1(x), \dots, z_p(x)]^T$ is a p-dimensional vector of synaptic weights called regressor vector and takes the form.

$$Z(x) = [z_1(\|x - c_1\|), \dots, z_p(\|x - c_n\|)] \quad (2.71)$$

Where, $z_i(\cdot)$ are the radial basis functions and $c_i (i=1, \dots, p)$ are distinct points in state space (termed centers). In this thesis the regressor terms involve radial basis functions (RBFs) with fixed centers and widths. A widely used RBF is the Gaussian function $z(x) = \exp\left(-\left\|\frac{x-c}{s}\right\|^2\right)$ where c and s are the center and the width of the receptive field respectively. It has been proven in [33] that an RBF network with sufficiently large node number n and appropriately placed node centers and variances can approximate any continuous function $g(x): \Omega_x \in \mathfrak{R}$ over a compact set $\Omega_x \subset \mathfrak{R}^n$ to arbitrary accuracy according to $g(x) = W^{*T} Z(x) + \varepsilon(x), \forall x \in \Omega_x$ (2.72) where W^* are the ideal constant weights and $\varepsilon(x)$ the approximation error in a way that:

$$\sup_{x \in \Omega_x} \{g(x) - W^{*T} Z(x)\} \leq \varepsilon$$

It is normally assumed that the ideal weight vector W^* exist such that $|\varepsilon(x)| < \varepsilon^*$ (with $\varepsilon^* > 0$) for all $x \in \Omega_x$. The ideal weight vector W^* is an artificial quantity required for analytical purposes and it is defined as the value of W that minimizes $|\varepsilon|$ for all $x \in \Omega_x \subset \mathfrak{R}^n$ that is:

$$W^* \triangleq \arg \min_{W \in \mathfrak{R}^n} \left\{ \sup_{x \in \Omega_x} |f(x) - W^T Z(x)| \right\} \quad (2.73)$$

So by the above equations it is concluded that if the number of regressors p is sufficiently large and the regressor terms are appropriately selected, then there exists a weight vector W^* such that W^{*T} can approximate $g(x)$ to any degree of accuracy in a given compact set Ω_x . The term $\sup_{x \in \Omega_x} \{g(x) - W^{*T} Z(x)\}$ becomes smaller as p increases, and unknown function $g(x)$ can be calculated as it is shown in the equation (2.72).

D. Persistency of Excitation

The property of Persistence of excitation is widely used in adaptive systems as it provides the conditions leading to the parameter convergence to the desired values. Especially in the field of RBF networks, persistency of excitation is widely used giving conditions under which the successful identification of nonlinear systems is achieved. It has been shown recently [34] that if the inputs to an RBF network coincide with the network neuron centers then the corresponding regressor vector satisfies the Persistency of Excitation condition. Also for RBF networks with neuron centers dispersed on a regular lattice, it was shown that the corresponding regressor vector is persistently excited provided that the input variables to the RBF networks belong to a certain neighborhood of neuron centers [35,11]. An important property which is applied in this thesis is shown in [32]. Especially when the regressor vector constructed out of RBF approximants is persistently exciting provided a kind of “ergodic” condition is satisfied. The size of the neighborhoods is restricted to be less than one half of the minimal distance between any two neuron centers and a class of ideal input trajectories which ensure the satisfaction of the PE condition are characterized as periodic or ergodic trajectories visiting the limited neighborhoods of all neuron centers of the RBF network.

Based on the papers [32,37] the relevant theoretical background of Persistency of Excitation which is used in this thesis is presented below.

The definition of the persistency of excitation which includes the discrete and continuous systems is the following:

Let μ be a positive, Σ -finite Borel measure on $[0, \infty)$. A continuous uniformly-bounded vector-valued function $Z: [0, \infty) \rightarrow \mathfrak{R}^n$ is persistently exciting, if there exist positive constants a_1, a_2 and T such that:

$$a_1 \|W\|^2 \leq \int_{t_0}^{t_0+T} |W^T Z(\tau)|^2 d\mu(\tau) \leq a_2 \|W\|^2 \quad (2.74)$$

And it holds for any $t_0 \geq 0$ and any constant vector $W \in \mathfrak{R}^p$. When RBFs are employed in solving the identification problem the vector $W(t)$ has the form:

$W(t) = (F(\|x(t) - c_1\|) \dots F(\|x(t) - c_N\|))^T$ (2.75), where F is the RBF, each c_i is a given point in state space, termed a center and $x(t)$ is the target trajectory. The function $x(t)$ is a continuous map from $[0, \infty)$ into $\mathfrak{R}^s$ the state space. Furthermore, we assume that $x(t)$ remains in a bounded

subset of $\mathfrak{R}^s$. The N centers $c_1, \dots, c_N$ are distinct points in $\mathfrak{R}^s$. It was shown that a sufficient condition for W of the form (2.14) to be persistently exciting amounts to an “ergodic” condition on $x(t)$, which is stated in the following theorem:

Theorem 1: Let $I = [t_0, t_0 + T]$ with $t_0, T > 0$ and $I_i = \{t \in I : \|x(t) - c_i\| \leq \epsilon\}, i = 1, \dots, p$ with $\epsilon < \frac{1}{2} \min_{i \neq j} \{\|c_i - c_j\|\}$. If there exists T such that $\mu(I_i), i = 1, \dots, p$ are bounded below by a positive constant τ_0 independent of t_0 and i , then $Z(x(t))$ is persistently exciting satisfying the equation (2.74).

From the above theorem 1 it is concluded that for a periodic orbit $x(t)$ with period T , if in each time interval $[t_0, t_0 + T]$ the periodic orbit $x(t)$ visits the ϵ -neighborhood of each center c_i of the RBF network for a minimum amount of time $\tau_0 > 0$ independent of t_0 where $\epsilon < \frac{1}{2} \min_{i \neq j} \{\|c_i - c_j\|\}$, then $Z(x(t))$ is persistently exciting. The restriction on ϵ guarantees that the ϵ -neighborhoods of the RBFs centers are non-intersecting so that for any bounded μ -measurable set I the subsets I_i defined in the previous theorem are disjoint and the following inequality holds for any constant vector $W \in \mathfrak{R}^p$.

E. Prescribed Performance

The control design is heavily connected to the prescribed performance notion that was originally employed to design robust controllers for various classes of nonlinear systems namely feedback linearizable [38], strict feedback [39,41], and general MIMO affine in the control [40], capable of guaranteeing output tracking with prescribed performance. In this thesis prescribed performance is restricted to tracking error convergence to a predefined arbitrarily small residual set. Paper [37] gives the basic theoretical background of prescribed performance in which the controller design is based and all this information is presenting below.

Consider a generic tracking error $e(t)$. Prescribed performance is achieved if $e(t)$ evolves strictly within a predefined that is bounded by decaying functions of time. The mathematical expression of prescribed performance is given $\forall t \geq 0$, by the following inequalities:

$$-\rho(t) < e(t) < \rho(t)$$

Where $\rho(t)$ is a smooth, bounded, strictly positive and decreasing function of time satisfying the condition $\lim_{t \rightarrow \infty} \rho(t) > 0$, called performance function [38]. The aforementioned statements are clearly illustrated in figure 1 for an exponentially decreasing performance function $\rho(t) = (\rho_0 - \rho_\infty)e^{-lt} + \rho_\infty$ with ρ_0, ρ_∞, l appropriately chosen strictly positive constants.

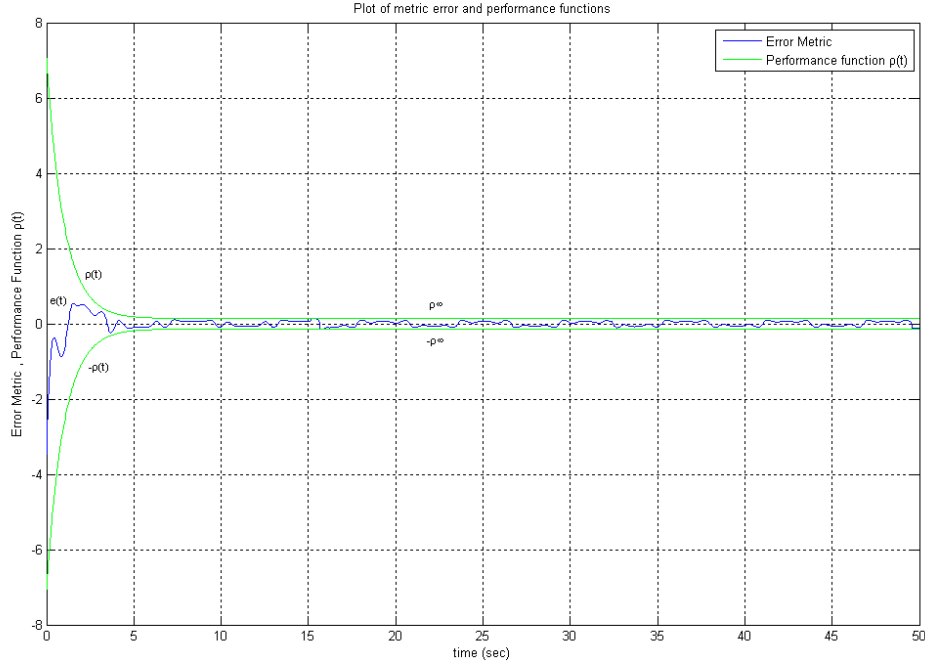


Figure 1. Graphical illustration of the prescribed performance definition

The constant $\rho_0 = \rho(0)$ is selected such that $\rho_0 > |e(0)|$. The constant $\rho_\infty = \lim_{t \rightarrow \infty} \rho(t)$ represents the maximum allowable size of the tracking error $e(t)$ at the steady state. Moreover, the decreasing rate of $\rho(t)$, which is affected by the constant l in this case, introduces a lower bound on the required speed of convergence of $e(t)$.

F. Dynamical Systems

Paper [37] gives basic theoretical background for dynamical systems and their properties which are used in the main results chapter of this thesis and they are presented below:

Consider the initial value problem:

$$\dot{\psi} = H(t, \psi), \psi(0) = \psi^0 \in \Omega_{\psi} \quad (2.76)$$

With $H: \mathfrak{R}_+ \times \Omega_{\psi} \rightarrow \mathfrak{R}^n$ where $\Omega_{\psi} \subset \mathfrak{R}^n$ is a non-empty open set.

Definition 1: A solution $\psi(t)$ of the initial value problem (2.76) is maximal if it has no proper right extension that is also a solution of (2.76).

As an example consider the initial value problem $\dot{\psi} = \psi^2$, $\psi(0) = 1$, whose solution is $\psi(t) = \frac{1}{1-t}$, $\forall t \in [0, 1)$. The solution is maximal since it cannot be defined for $t > 1$. Stated otherwise, there is no proper extension of $\psi(t)$ to the right of $t = 1$ that is also a solution of the original initial value problem.

Theorem 2: Consider the initial value problem (2.76). Assume that $H(t, \psi)$ is a) locally Lipschitz on ψ for almost all $t \in \mathfrak{R}_+$ b) piecewise continuous on t for each fixed $\psi \in \Omega_{\psi}$ and c) locally integrable on t for each fixed $\psi \in \Omega_{\psi}$. Then, there exists a maximal solution $\psi(t)$ of (2.76) on the time interval $[0, \tau_{\max})$ with $\tau_{\max} > 0$ such that $\psi(t) \in \Omega_{\psi}$, $\forall t \in [0, \tau_{\max})$.

Proposition 1: Assume that the hypotheses of Theorem 2 hold. For a maximal solution $\psi(t)$ on the timer interval $[0, \tau_{\max})$ with $\tau_{\max} < \infty$ and for any compact set $\Omega'_{\psi} \subset \Omega_{\psi}$ there exists a time instant $t' \in [0, \tau_{\max})$ such that $\psi(t') \notin \Omega'_{\psi}$.

CHAPTER 3 – MAIN RESULTS

In this part firstly a technique which is producing a reference trajectory is analyzed and designed in such a way to satisfy the persistency of excitation condition a priori. Secondly a control design is followed based on the prescribed performance succeeding in making the system states follow the desired trajectory and satisfying the persistency of excitation of the RBF neural network regressors that are utilized to achieve local learning of system nonlinear functions. Furthermore an identification scheme is designed to achieve the aim of learning in a predefined desired compact set of system states.

A. Reference design

Reference trajectory is aimed to be designed in such a way as satisfying the persistency of excitation condition a priori. As stated in [34] if the inputs to RBF coincide with the neuron centers placed in a lattice inside a predefined compact set then the regressor vector of neural network applied satisfies the persistency of excitation and the estimates of weight vectors converges to their optimal values satisfying the learning purpose.

The predefined compact set, is a bounded rectangle for which x-coordinates represent the values of position of the system, and y-coordinates represent the velocity values accordingly. So, this rectangle represents all possible states of the system inside this area. This set is defined as the desired area inside which, the aim of learning is needed to be achieved. This kind of learning method is characterized as local learning method. In this compact set a grid is created by placing the nodes of the grid in a way to cover as much as possible all the possible states of system functionality. These nodes are the neuron centers of Radial Basis Functions (RBFs). Increasing the number of nodes improves the result of learning but it also increases the computational needs. An example of this compact set with the nodes chosen is appeared in the following figure. In this case 36 nodes are chosen to construct the lattice of the compact set.

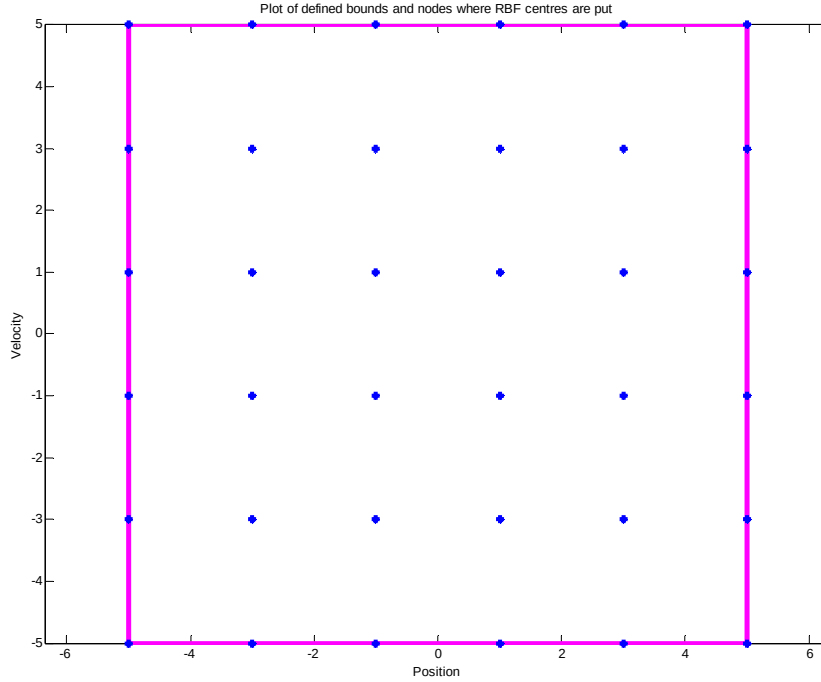


Figure 2. Compact set with the grid of the selected nodes

After the placing the nodes and creating the grid, next step is to define the desired trajectory which is designed to traverse all the nodes of the grid in order to satisfy the persistency of excitation. As x-coordinates and y-coordinates represent system position values and velocities as well, the trajectory is designed in a way as its x-coordinates be represented by $x_{ref}(t)$ and satisfy the following equation:

$$x_{ref}(t) = A \sin(\omega t + \varphi) + A_0 \quad (3.1)$$

Also, as y-coordinates of the compact set represent the velocity values, then the y-coordinates of the desired trajectory are given from the differentiation of the above equation(3.1).

$$y_{ref}(t) = \dot{x}_{ref}(t) = A\omega \cos(\omega t + \varphi) \quad (3.2)$$

As it is $\sin^2(\omega t + \varphi) + \cos^2(\omega t + \varphi) = 1$, from the above equations (3.1),(3.2) it is derived:

$$\frac{(x_{ref}(t) - A_0)^2}{A^2} + \frac{(\dot{x}_{ref}(t))^2}{(A\omega)^2} = 1 \quad (3.3)$$

The above equation is an ellipse with center $(A_0, 0)$ and A and $A\omega$ are the lengths of ellipses major and minor axis and the ellipse passes from the point $(x_{ref}(t), \dot{x}_{ref}(t))$ which belongs in the compact set. The meaning of the equation (3.3) is that for any point inside the predefined set, an ellipse can be found which passes at least through this point having zero y-coordinates in its center and x-coordinates of its center must be inside the set. The first aim is to construct such a separate ellipse to pass through each node (where the neuron centers are placed) in the desired

compact set. Secondly it is preferred that each ellipse's trajectory be as much as possible inside the compact set traversing apart from the basic node from which is designed to pass, as many other nodes (or in a close area to them) as possible improving the final result of learning and reducing the time needed for the learning process. It is concluded that for every node an optimal ellipse must be found with certain parameters satisfying the goals and constraints mentioned previously. This is an optimization problem, thus an optimization algorithm is designed which is analyzed next.

The problem defined above is to find the optimal ellipse passing through the given node inside a compact set following an ellipse trajectory where the center of the ellipse have zero y-coordinate and x-coordinate inside the compact set. The goal needed to be satisfied is to run as long distance (largest possible perimeter) as possible inside the compact set. So, the center of the ellipse needed to be calculated is in the form $C(x_0, 0)$. Assuming as $\bar{x}, -\bar{x}$ the upper and lower bounds in x-axis of the compact set, and $\bar{y}, -\bar{y}$ are the upper and lower bounds in y-axis correspondingly. So for x_0 it holds that: $-\bar{x} \leq x_0 \leq \bar{x}$. Then we assume a random (but given) point $P(P_x, P_y)$ inside the compact set, through which the ellipse must pass. Two kinds of ellipses could pass through this given point as shown in the following figure:

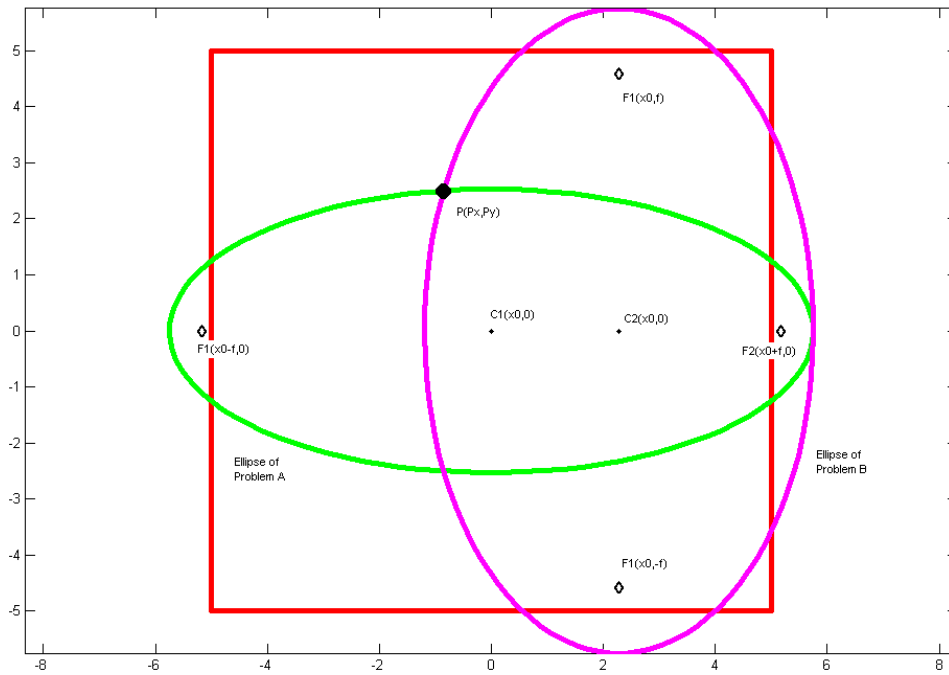


Figure 3. Optimal ellipses passing from a given point inside a compact set

The difference between these two kinds of ellipses is that the focal points of first kind of ellipse have zero y-coordinates being placed on x-axis and the distance of the major axis distance a is running across x -axis, while in the second kind of ellipse the two focal points are placed bilaterally in symmetric distances from the x-axis and the major axis distance a is running across the y -axis. So, it is concluded that two optimization problems must be solved, where each time an optimal ellipse from each of the two kinds shown in the above figure 3 is calculated.

Problem A: For the first kind of ellipse it holds for the focal points' F_1, F_2 coordinates:

$$F_1(x_0 - f, 0), F_2(x_0 + f, 0)$$

Where x_0 the x-coordinate of the ellipse center and f the distance between the two focal points. Each ellipse has two perpendicular axes about which the ellipse is symmetric. These axes intersect at the center of the ellipse due to this symmetry. The larger of these two axes, which corresponds to the largest distance between antipodal points on the ellipse, is called major axis, and this distance is expressed by the quantity $2a$ where a the distance of the one of the two semi-axes. The smaller of the ellipse's axes and the smallest distance across the ellipse, is called minor axis. This distance is expressed by the quantity $2b$ where b the distance of the each one of the two semi-axes. Distances a, b are shown in the figure 3.

It holds for each ellipse that the sum of distances from any point $P(P_x, P_y)$ on the ellipse to those two focal points is constant and equal to the major axis. This property for the first kind of ellipse is expressed by the following equation:

$$(PF_1) + (PF_2) = 2a \Rightarrow \sqrt{[P_x - (x_0 - f)]^2 + P_y^2} + \sqrt{[P_x - (x_0 + f)]^2 + P_y^2} = 2a \quad (3.4)$$

Another property is that, the distance from the center C to either of the focal points is $f = ae$ (3.5), where a the major axis and e the eccentricity of the ellipse. The distance f can be expressed in terms of the minor and major axis distances due to the Pythagorean Theorem as:

$$f = \sqrt{a^2 - b^2} \Rightarrow b = \sqrt{a^2 - f^2} \quad (3.6)$$

At next step the objective function of the optimization problem A should be defined as:

$$V(a, b) = -w_1 ab + w_2 f \quad (3.7)$$

The goal of the optimization problem is to maximize the trajectory of each ellipse inside the area of the compact set meaning to maximize the perimeter of the ellipse. The perimeter of an ellipse is very difficult to be calculated and there is not any analytical equation for this, just only approximations like the following equation: $p \approx 2\pi \sqrt{\frac{a^2 + b^2}{2}}$, where p the ellipse's perimeter and a, b the major and minor semi-axis distances. The maximization of the ellipse's perimeter

can be also achieved by maximizing the area of the ellipse given from the equation: $A_{\text{ellipse}} = \pi ab$. Also another goal is to minimize the distance between the two focal points f as we want to keep as much as possible the trajectory of the ellipse inside the compact set. So the optimization problem can be defined by minimizing the quantity $-w_1 ab + w_2 f$, where w_1, w_2 are the weight factors expressing the importance of the optimization goals. In our approach it is more important to maximize the area of the ellipse, so we use $w_1 = 2$ and $w_2 = 1$ correspondingly. So the optimization goal can be written as: $\min\{V(a, b)\}$.

Next step in the optimization problem is to define the constraints. As the ellipse is needed to be kept inside the given compact set the bounds of the ellipse in each direction must be inside the compact set correspondingly. So for the lower ellipse point on the left having the coordinates $(x_0 - a, 0)$ and the higher ellipse point on the right $(x_0 + a, 0)$ it holds as a constraint:

$$x_0 - a > -\bar{x} \text{ and } x_0 + a < \bar{x} \quad (3.8)$$

The above constraints are equal to the following one:

$$|x_0| < \bar{x} - a \quad (3.9)$$

Also a-axis and b-axis of the ellipse must be confined within the compact set which is represented by the constraints:

$$a < \bar{x} \quad (3.10) \text{ and } b < \bar{y} \quad (3.11)$$

As constraining the ellipse's trajectory to be in its whole inside the compact set, this makes the optimization algorithm calculate an ellipse (passing from the given node) with short perimeter distance. If we make less strict the constraints confining the ellipse's trajectory inside the compact set then the perimeter of the optimal ellipse would be maximized. This can be done allowing a certain percentage of the ellipse's trajectory to be crossed outside the compact set by defining it with a certain parameter. So, the constraints (3.9), (3.10), (3.11) can be transformed into:

$(3.9) \Rightarrow |x_0| < D\bar{x} - a \quad (3.12)$ where $D \geq 1$ a coefficient which moves the bound of the compact set further from $\bar{x}$ in both directions right and left from the compact set as well. Also, $(3.10) \Rightarrow b < D_d \bar{x} \quad (3.13)$ where $D_d \geq 1$ a coefficient which transposes the bound of the compact set further from $\bar{x}$ in both directions above and below the compact set. Also for (3.11) it holds that:

$$(3.11) \Rightarrow a < D\bar{x} \quad (3.14)$$

Also, the center of the ellipse must be inside the compact set, so the following constraints are defined $x_0 < \bar{x}$ and $x_0 > -\bar{x}$ which are equal to:

$$|x_0| < \bar{x} \quad (3.15)$$

Finally it should be ensured that lines are excluded and that optimization algorithm each time gives an ellipse shape as a solution. This can be ensured by constraining ellipse's axis b not to be

zero value. So, if $f = a$ then from the equation (3.6) it is $b = 0$. So, by confining the distance between the two focal points f to be a bit less than a then it can be ensured that $b > 0$ and this can be done by putting:

$$|f| \leq 0.96a \Rightarrow -0.96a \leq f \leq 0.96a \quad (3.16)$$

So, in summary the optimization problem A has been fully defined with the objective function to be minimized:

$$V(a, b) = -w_1 ab + w_2 f \quad (3.17)$$

$$\text{Optimization goal: } \min\{V(a, b)\}$$

And the constraints of the problem are:

$$|x_0| < D\bar{x} - a \quad (3.18), \quad D \geq 1$$

$$b < D_d \dot{\bar{x}} \quad (3.19), \quad D_d \geq 1$$

$$a < D\bar{x} \quad (3.20)$$

$$|x_0| < \bar{x} \quad (3.21)$$

$$|f| \leq 0.96a \quad (3.22)$$

Problem B: The same procedure is followed in the problem B, in which a different kind of ellipse is calculated as it is shown in the figure 3 in blue color. For the second kind of ellipse it holds for the focal points' F_1, F_2 coordinates:

$$F_1(x_0, -f), \quad F_2(x_0, f)$$

Also the same property as defined in equation (3.4) holds:

$$(PF_1) + (PF_2) = 2a \Rightarrow \sqrt{[(P_x - x_0)^2 + (P_y + f)^2]} + \sqrt{[(P_x - x_0)^2 + (P_y - f)^2]} = 2a \quad (3.23)$$

It also holds that:

$$f = \sqrt{a^2 - b^2} \Rightarrow b = \sqrt{a^2 - f^2} \quad (3.24)$$

The objective function is of the same form as defined in the problem A with the same minimization goals.

$$V(a, b) = -w_1 ab + w_2 f \quad (3.25)$$

The optimization goal can be written as: $\min\{V(a, b)\}$. Next step is the definition of the constraints. In the same way as the ellipse is needed to be kept inside the given compact set the bounds of the ellipse in each direction must be inside the compact set correspondingly. So for the lower ellipse point on the left having the coordinates $(x_0 - b, 0)$ and the higher ellipse point on the right $(x_0 + b, 0)$ it holds as a constraint:

$$x_0 - b > -\bar{x} \quad \text{and} \quad x_0 + b < \bar{x}$$

The above constraints are equal with the following one:

$$|x_o| < \bar{x} - b \quad (3.26)$$

A parameter is introduced in the above constraint to make less strict this constraint and allow a certain percentage of the ellipse's trajectory to be crossed outside the compact set. Then the above constraint becomes:

$$|x_o| < D\bar{x} - b \quad (3.27), \quad D \geq 1$$

Also x-axis and b-axis of the ellipse must be confined within the compact set which is represented by the following constraints after introducing the relevant parameters D, D_d which allow a certain percentage of the ellipse's trajectory to be crossed outside the compact set:

$$b < D\bar{x} \quad (3.28)$$

$$a < D_d \dot{\bar{x}} \quad (3.29)$$

Also, the center of the ellipse must be inside the compact set, so the following constraints are defined $x_o < \bar{x}$ and $x_o > -\bar{x}$ which are equal to:

$$|x_o| < \bar{x} \quad (3.30)$$

So, in summary the optimization problem A has been fully defined with the objective function to be minimized:

$$V(a, b) = -w_1 ab + w_2 f \quad (3.31)$$

$$\text{Optimization goal: } \min\{V(a, b)\}$$

And the constraints of the problem are:

$$|x_o| < D\bar{x} - b \quad (3.32), \quad D \geq 1$$

$$b < D\bar{x} \quad (3.33)$$

$$a < D_d \dot{\bar{x}} \quad (3.34)$$

$$|x_o| < \bar{x} \quad (3.35)$$

From each optimization problem a different ellipse is produced and next it is chosen the one which has the minimum value of the objective function having the largest area as well as the largest perimeter inside the compact set.

So, after calculation of the optimal ellipse for each given node is done, then a set of ellipses is produced where each ellipse passes through at least one node. Also the four edges points of the compact set are regarded as nodes and a separate ellipse for each one is calculated as well. For this case of finding the optimal ellipse passing from the edges, the optimization problem written above is not well defined and greater percentage of the optimal ellipse's trajectory should be allowed to be crossed outside the compact set so as a feasible solution to be found. An example of

all ellipses from which each one is calculated passing through at least one of the 9 selected nodes and their plot inside a given compact set is shown in the following figure:

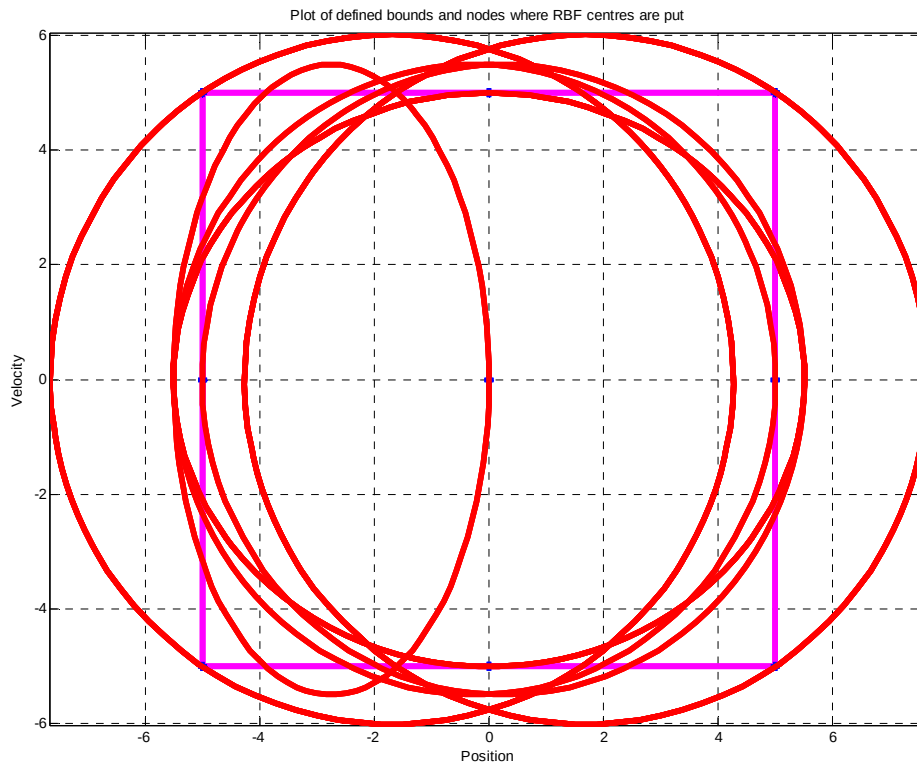


Figure. 4 Plot of ellipses calculated from the optimization algorithm and for each node there is one ellipse passing through it inside the compact set

Also, it has been mentioned previously that the neuron centers of Radial Basis Functions (RBFs) are put in the points where the nodes of the lattice has been placed. An example of this is given in the following 3D plot where the compact set is appeared with the nodes and the ellipses calculated for each node. Also the radial basis functions are also appeared whose neuron centers are placed in the nodes of the lattice.

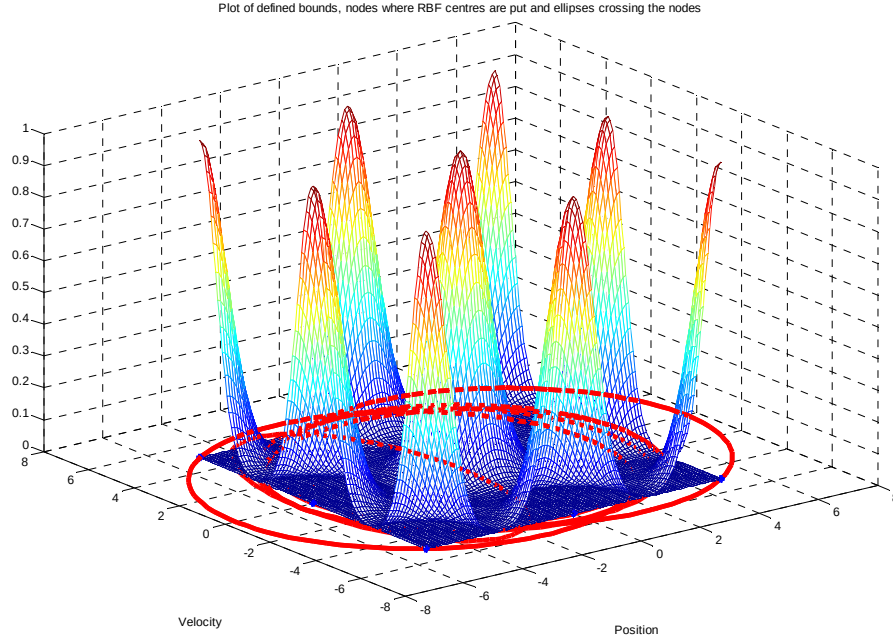


Figure 5. 3D Plot of compact set with nodes, ellipses passing through them and Radial Basis Functions

Next we assume a route which starts from the top left node following the whole trajectory of the first ellipse two times and next doing the same by moving onto the trajectory of the next consecutive node's ellipse on the right until reaching the last node. The same movement is done from the last node moving again back to the first node. This is the complete route which can be repeated periodically during the learning process. The desired reference trajectory is defined from the following equations:

$$x_{ref}(t) = A_i \sin(\omega_i t + \varphi_i) + A_{0i} \quad (3.36)$$

$$\dot{x}_{ref}(t) = A_i \omega_i \cos(\omega_i t + \varphi_i) \quad (3.37)$$

Where, $A_i, \omega_i, \varphi_i, A_{0i}$ are the parameters of each ellipse's trajectory of the whole route. So the whole desired reference trajectory for the learning problem is calculated from the above equations(3.36), (3.37) by changing parameters $A_i, \omega_i, \varphi_i, A_{0i}$ as moving from one ellipse's trajectory to the next. Each ellipse's trajectory is the optimal one calculated for each node from which the ellipse passes through. The route to be followed is formed by running the whole ellipses trajectories starting from the trajectory corresponding to the first node till the one corresponding to the final node and then doing the opposite direction as well. Each time, the nodes which correspond to each ellipse's trajectory must be neighboring keeping a consecutive order during the move from the one node's ellipse to the next. As the transition from the one ellipse's trajectory to the consecutive one is regarded to be done at zero time, the reference

trajectory $(x_{ref}, \dot{x}_{ref})$ is filtered in order to smooth the regions of abrupt transition where discontinuities are found.

B. Control design

As the reference trajectory is defined, a controller is designed to force system to follow this desired trajectory for the aim of learning to be achieved. An adaptive controller is designed using the prescribed performance method analyzed in the chapter 2-B and the Lyapunov stability analysis.

After given the compact set Ω_x where a locally accurate approximation of the unknown functions $f(\cdot), g(\cdot)$ to be achieved and also a smooth bounded trajectory $x_{ref}(t)$ defined in the chapter 3-A and satisfying $\bar{x}_{ref}(t) = [x_{ref}(t), \dot{x}_{ref}(t)]^T \in \Omega_x, \forall t > 0$ and any initial system condition $x(0) \in \Omega_x$, we define the state errors:

$$e_1 = x_1 - x_d \quad (3.38)$$

$$e_2 = x_2 - \dot{x}_d \quad (3.39)$$

Where, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ the overall state vector and $x_1 = [x_1, \dots, x_n]^T$ the vector of positions and

$x_2 = [\dot{x}_1, \dots, \dot{x}_n]^T$ the vector of velocities of the Lagrange system. $x_d = \begin{bmatrix} x_{ref}(t) \\ \cdot \\ \cdot \\ \cdot \\ x_{ref}(t) \end{bmatrix}$ the reference signal

that position states of each degree of freedom will follow, and $\dot{x}_d = \begin{bmatrix} \dot{x}_{ref}(t) \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_{ref}(t) \end{bmatrix}$ the reference signal

that velocity states of each degree of freedom will follow.

Also, $\underline{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$ the overall error vector. Then we calculate the error metric which is in generic

form: $\underline{\varepsilon}(\underline{e}(t)) = \left(\frac{d}{dt} + \lambda \right)^{n-1} e_1(t) = \sum_{i=0}^{n-1} \binom{n-1}{i} \underline{\lambda}^i \underline{e}_{n-i}(t)$ (3.40), where n represents the states of the system. In the case examined in this thesis $n = 2$, so the metric error is:

$$\underline{\varepsilon}(\underline{e}(t)) = \sum_{i=0}^1 \binom{1}{i} \underline{\lambda}^i \underline{e}_{2-i}(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \underline{\lambda}^0 \underline{e}_2(t) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \underline{\lambda}^1 \underline{e}_1(t) = \underline{e}_2(t) + \underline{\lambda} \underline{e}_1(t) \quad (3.41)$$

Where $\underline{\lambda} = \begin{bmatrix} \lambda_1 & 0 & . & . & . & 0 \\ 0 & \lambda_1 & 0 & . & . & . \\ . & 0 & . & 0 & . & . \\ . & . & 0 & . & 0 & . \\ . & . & . & 0 & . & 0 \\ 0 & . & . & . & 0 & \lambda_n \end{bmatrix}$, $\lambda_1, \lambda_2, \dots, \lambda_n > 0$ (3.42), is a control gain matrix. At next step a

performance function $\underline{\rho}_{\varepsilon}(t)$ has to be defined satisfying the condition $\underline{\rho}_{\varepsilon}(0) > |\underline{\varepsilon}(\underline{e}(0))|$ (3.43). Performance function as defined in the chapter 2-B and for a Lagrangian system of 2nd order with N degrees of freedom, performance function is:

$$\underline{\rho}_{\varepsilon}(t) = \text{diag} \{ \rho_{i\varepsilon}(t) \} = \begin{bmatrix} \rho_{1\varepsilon}(t) & 0 & . & . & . & 0 \\ 0 & \rho_{2\varepsilon}(t) & 0 & . & . & 0 \\ . & 0 & . & 0 & . & . \\ . & . & 0 & . & 0 & . \\ . & . & . & 0 & . & 0 \\ 0 & . & . & . & 0 & \rho_{n\varepsilon}(t) \end{bmatrix} \quad (3.44), \text{ where } n \text{ is the number of degrees}$$

of freedom. Also, the condition(3.43), takes the matrix form: $\begin{bmatrix} \rho_{1\varepsilon}(0) > |\varepsilon_1(0)| \\ \rho_{2\varepsilon}(0) > |\varepsilon_2(0)| \\ . \\ . \\ \rho_{n\varepsilon}(0) > |\varepsilon_n(0)| \end{bmatrix}$ (3.45). Taking

the above equations into account, the following **control law** is designed:

$$u = -K \cdot \text{diag} \left\{ \frac{2}{1 - \xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \quad (3.46)$$

Where, K the control gain matrix:

$$K = \begin{bmatrix} k_1 & 0 & . & . & . & 0 \\ 0 & k_2 & 0 & . & . & . \\ . & 0 & . & 0 & . & . \\ . & . & 0 & . & 0 & . \\ . & . & . & 0 & . & 0 \\ 0 & . & . & . & 0 & k_n \end{bmatrix} \quad (3.47) \text{ and } \xi_\varepsilon = \begin{bmatrix} \xi_{\varepsilon_1} \\ \xi_{\varepsilon_2} \\ . \\ . \\ . \\ \xi_{\varepsilon_n} \end{bmatrix} \quad (3.48) \text{ the normalized error defined as:}$$

$$\underline{\xi}_\varepsilon = \underline{\rho}_\varepsilon^{-1}(t) \underline{\varepsilon}(\underline{e}(t)) = \underline{\rho}_\varepsilon^{-1}(t) [\underline{e}_2(t) + \underline{\lambda} \underline{e}_1(t)] \quad (3.49) \text{ and } \underline{S}_\varepsilon = \begin{bmatrix} S_{\varepsilon_1} \\ S_{\varepsilon_2} \\ . \\ . \\ . \\ S_{\varepsilon_n} \end{bmatrix} \quad (3.50) \text{ a matrix of signals } S_{\varepsilon_i}$$

defined as: $S_{\varepsilon_i} = \ln \left(\frac{1 + \xi_{\varepsilon_i}}{1 - \xi_{\varepsilon_i}} \right)$ (3.51). The proof of how the equation (3.46) is derived is presented analytically in the following theorem.

Theorem 3: Consider system (2.62) obeying assumptions 1 and 2. Given a compact set Ω_x , any initial system conditions $\underline{x}(0) \in \Omega_x$ and any smooth bounded trajectory $\bar{x}_{ref}(t) \in \Omega_x$, the control input (3.46), guarantees that: i) all the signals in the closed loop remain bounded $\forall t \geq 0$ and additionally ii) the state errors $\underline{e} = \begin{bmatrix} \underline{e}_1 \\ \underline{e}_2 \end{bmatrix}$, where $\underline{e}_1 = \underline{x}_1 - \underline{x}_d$ and $\underline{e}_2 = \underline{x}_2 - \dot{\underline{x}}_d$, converge in finite time to the compact sets:

$$E_1 := \left\{ \underline{e}_1 \triangleq [e_1^1, e_1^2, \dots, e_1^n] \in \mathfrak{R}^n : |e_1^j| \leq \frac{\lim_{t \rightarrow \infty} \rho_{\varepsilon_j}(t)}{\lambda_j} \right\}, \quad E_2 := \left\{ \underline{e}_2 \triangleq [e_2^1, e_2^2, \dots, e_2^n] \in \mathfrak{R}^n : |e_2^j| \leq 2 \lim_{t \rightarrow \infty} \rho_{\varepsilon_j}(t) \right\} \text{ for } j = 1, \dots, n, \text{ where } n \text{ the degrees of freedom of the Lagrange system.}$$

Proof: Differentiating equations (3.38) and (3.39) and as $\dot{\underline{x}}_1 = \underline{x}_2$ we take:

$$\dot{\underline{e}}_1 = \dot{\underline{x}}_1 - \dot{\underline{x}}_d = \underline{x}_2 - \dot{\underline{x}}_d = \underline{e}_2 \quad (3.52)$$

$$\dot{\underline{e}}_2 = \dot{\underline{x}}_2 - \ddot{\underline{x}}_d \quad (3.53)$$

Using equation (2.62) the above equation becomes:

$$(3.53) \Rightarrow \underline{f}(x) + \underline{g}(x) \underline{u} - \ddot{\underline{x}}_d \quad (3.54)$$

From (3.53), (2.64), (2.65) it is arisen:

$$\dot{\underline{e}}_2 = -M^{-1}(C\underline{x}_2 - G + \underline{u}) - \ddot{\underline{x}}_d \quad (3.55)$$

Differentiating (3.41) with respect to time it is arisen:

$$\dot{\underline{e}}(\underline{e}(t)) = \dot{\underline{e}}_2(t) + \underline{\lambda}\dot{\underline{e}}_1(t) = -M^{-1}(C\underline{x}_2 - G + \underline{u}) - \ddot{\underline{x}}_d + \underline{\lambda}\underline{e}_2(t) \quad (3.56)$$

We define the normalized error:

$$\underline{\xi}_\varepsilon = \underline{\rho}_\varepsilon^{-1}(t)\underline{\varepsilon}(\underline{e}(t)) = \underline{\rho}_\varepsilon^{-1}(t)[\underline{e}_2(t) + \underline{\lambda}\underline{e}_1(t)] \quad (3.57)$$

Where $\underline{\rho}_\varepsilon(t)$ the performance functions matrix as defined in equation(3.44). The performance function parameters have been selected to satisfy the condition in equation(3.45). Next, the generalized error vector is defined:

$$\underline{\xi} = \begin{bmatrix} \underline{e} \\ \underline{\xi}_\varepsilon \end{bmatrix} \quad (3.58)$$

Differentiating (3.58) with respect to time we take:

$$\dot{\underline{\xi}} = h(t, \underline{\xi}) = \begin{bmatrix} \dot{\underline{e}} \\ \dot{\underline{\xi}}_\varepsilon \end{bmatrix} = \begin{bmatrix} \underline{e}_2 \\ -M^{-1}(C\underline{x}_2 - G + \underline{u}) - \ddot{\underline{x}}_d \\ (\underline{\rho}_\varepsilon^{-1}(t))' \underline{\varepsilon}(\underline{e}(t)) + \underline{\rho}_\varepsilon^{-1}(t)\dot{\underline{\varepsilon}}(\underline{e}(t)) \end{bmatrix} \quad (3.59)$$

Substituting the control input from equation(3.46), to the above equation (3.59) it might be written as:

$$\dot{\underline{\xi}} = h(t, \underline{\xi}) = \begin{bmatrix} \dot{\underline{e}} \\ \dot{\underline{\xi}}_\varepsilon \end{bmatrix} = \begin{bmatrix} \underline{e}_2 \\ -M^{-1} \left(C\underline{x}_2 - G + -K \cdot \text{diag} \left\{ \frac{2}{1 - \xi_{\varepsilon_i}^2} \right\} \underline{\rho}_\varepsilon^{-1}(t) \underline{S}_\varepsilon \right) - \ddot{\underline{x}}_d \\ (\underline{\rho}_\varepsilon^{-1}(t))' \underline{\varepsilon}(\underline{e}(t)) + \underline{\rho}_\varepsilon^{-1}(t)\dot{\underline{\varepsilon}}(\underline{e}(t)) \end{bmatrix} \quad (3.60)$$

Substituting in the above equation (3.60) the equations (3.39), (3.50), (3.51), (3.56), (3.57) and the control design (3.46), the above equation (3.60) might be written in the form:

$$\dot{\underline{\xi}} = h(t, \underline{\xi}) = \begin{bmatrix} \dot{\underline{e}} \\ \dot{\underline{\xi}}_\varepsilon \end{bmatrix} = \begin{bmatrix} \underline{W}_1 \\ \underline{W}_2 \\ \underline{W}_3 \end{bmatrix} \quad (3.61)$$

Where $\underline{W}_1, \underline{W}_2, \underline{W}_3$ are:

$$\underline{W}_1 = \underline{e}_2 \quad (3.62)$$

$$\underline{W}_2 = -M^{-1} \left(C(\underline{e}_2 + \dot{\underline{x}}_d) - G + -K \cdot \text{diag} \left\{ \frac{2}{1 - \xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \begin{bmatrix} \ln \left(\frac{1 + \xi_{\varepsilon_1}}{1 - \xi_{\varepsilon_1}} \right) \\ \ln \left(\frac{1 + \xi_{\varepsilon_2}}{1 - \xi_{\varepsilon_2}} \right) \\ \vdots \\ \ln \left(\frac{1 + \xi_{\varepsilon_n}}{1 - \xi_{\varepsilon_n}} \right) \end{bmatrix} \right) - \ddot{\underline{x}}_d \quad (3.63)$$

$$\underline{W}_3 = \left(\underline{\rho}_{\varepsilon}^{-1}(t) \right)' \underline{\rho}_{\varepsilon}(t) \underline{\xi}_{\varepsilon} + \underline{\rho}_{\varepsilon}^{-1}(t) \left(-M^{-1} \left(C(\underline{e}_2 + \dot{\underline{x}}_d) - G - K \cdot \text{diag} \left\{ \frac{2}{1 - \xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \begin{bmatrix} \ln \left(\frac{1 + \xi_{\varepsilon_1}}{1 - \xi_{\varepsilon_1}} \right) \\ \ln \left(\frac{1 + \xi_{\varepsilon_2}}{1 - \xi_{\varepsilon_2}} \right) \\ \vdots \\ \ln \left(\frac{1 + \xi_{\varepsilon_n}}{1 - \xi_{\varepsilon_n}} \right) \end{bmatrix} \right) - \ddot{\underline{x}}_d + \underline{\lambda} \underline{e}_2(t) \right) \quad (3.64)$$

In this point the open set $\Omega_{\xi} = \mathfrak{R}^n \times (-1, 1)$ is defined. In the sequel we proceed to two phases. First, the existence and uniqueness of a maximal solution $\xi(t)$ of (3.61) over the set Ω_{ξ} for a time interval $[0, \tau_{\max})$ (i.e., $\xi(t) \in \Omega_{\xi}, \forall t \in [0, \tau_{\max})$) is ensured. Secondly we prove that the proposed control scheme (3.46) guarantees, for all $t \in [0, \tau_{\max})$ a) the boundness of all closed loop signals of (3.61), as well as that b) $\xi(t)$ remains strictly within a compact subset of Ω_{ξ} , which leads by contradiction to $\tau_{\max} = \infty$ and consequently to the completion of the proof.

Part A. The set Ω_{ξ} is nonempty and open. Moreover, the performance function $\underline{\rho}_{\varepsilon}(t)$ has been selected to satisfy the condition $\underline{\rho}_{\varepsilon}(0) > |\varepsilon(e(0))|$ as expressed by equation (3.45). As a consequence, $|\underline{\xi}_{\varepsilon}(0)| < 1$ for all elements of the vector in equation (3.48). This results in $\underline{\xi}(0) \in \Omega_{\xi}$ since it is assumed that $\underline{x}(0), \underline{x}_d(0) \in \Omega_x$. Additionally, h is continuous on t and continuous on ξ over the set Ω_{ξ} . Therefore the hypotheses of Theorem 2 stated in the subsection

2-C holds and the existence and uniqueness of a maximal solution $\xi(t)$ of (3.61) on a time interval $[0, \tau_{\max})$ such that $\xi(t) \in \Omega_\xi, \forall t \in [0, \tau_{\max})$ is ensured.

Part B. It has been proven in Part A that $\xi(t) \in \Omega_\xi, \forall t \in [0, \tau_{\max})$ and more specifically that:

$$\xi(t) \in (-1, 1), \forall t \in [0, \tau_{\max}) \quad (3.65)$$

Utilizing the equation (3.49) and (3.65) we obtain that $\underline{\varepsilon}(t) = \underline{\xi}_\varepsilon(t) \underline{\rho}_\varepsilon(t)$ is absolutely bounded by $\underline{\rho}_\varepsilon(0)$ for all $t \in [0, \tau_{\max})$. So, equation (3.40) represents a stable linear system with input ε and output e . Hence, $e_1(t)$ and (its derivatives up to $n-1=2-1=1$) its first derivative are bounded. As a result, there exists a compact set $\Omega_e \subset \mathcal{R}^n$, the size of which is independent of $\tau_{\max}$, such that $e(t) \in \Omega_e, \forall t \in [0, \tau_{\max})$. Furthermore, the following signals are well defined for all $t \in [0, \tau_{\max})$.

$$\underline{S}_e(t) = \begin{bmatrix} \ln\left(\frac{1+\underline{\xi}_{\varepsilon_1}(t)}{1-\underline{\xi}_{\varepsilon_1}(t)}\right) \\ \ln\left(\frac{1+\underline{\xi}_{\varepsilon_2}(t)}{1-\underline{\xi}_{\varepsilon_2}(t)}\right) \\ \vdots \\ \ln\left(\frac{1+\underline{\xi}_{\varepsilon_n}(t)}{1-\underline{\xi}_{\varepsilon_n}(t)}\right) \end{bmatrix} \quad (3.66)$$

Next, we consider the Lyapunov function candidate:

$$V = \frac{1}{2} \underline{S}_e^T \underline{K}^{-1} \underline{S}_e \quad (3.67)$$

Where $\underline{K}$ the controller gain matrix as defined in (3.47). Differentiating with respect to time the positive definite and radially unbounded function V , we obtain:

$$\dot{V} = \underline{S}_e^T \underline{K}^{-1} \dot{\underline{S}}_e \quad (3.68)$$

Differentiating the quantities $\underline{S}_e, \underline{\xi}_\varepsilon$ we obtain:

$$\dot{\underline{S}}_e = \begin{bmatrix} \dot{S}_{\varepsilon_1} \\ \dot{S}_{\varepsilon_2} \\ \vdots \\ \dot{S}_{\varepsilon_n} \end{bmatrix} \quad (3.69), \quad \dot{\underline{\xi}}_e = \begin{bmatrix} \dot{\xi}_{\varepsilon_1} \\ \dot{\xi}_{\varepsilon_2} \\ \vdots \\ \dot{\xi}_{\varepsilon_n} \end{bmatrix} \quad (3.70)$$

From equation (3.51) we obtain:

$$\dot{S}_{\varepsilon_i} = \frac{dS_{\varepsilon_i}}{d\xi_{\varepsilon_i}} \cdot \frac{d\xi_{\varepsilon_i}}{dt} = \frac{1-\xi_{\varepsilon_i}}{1+\xi_{\varepsilon_i}} \cdot \frac{(1+\xi_{\varepsilon_i})'(1-\xi_{\varepsilon_i}) - (1+\xi_{\varepsilon_i})(1-\xi_{\varepsilon_i})'}{(1-\xi_{\varepsilon_i})^2} \cdot \dot{\xi}_{\varepsilon_i} = \frac{1-\xi_{\varepsilon_i} + 1+\xi_{\varepsilon_i}}{1-\xi_{\varepsilon_i}^2} \cdot \dot{\xi}_{\varepsilon_i} = \frac{2}{1-\xi_{\varepsilon_i}^2} \cdot \dot{\xi}_{\varepsilon_i} \quad (3.71)$$

From equation we (3.57) obtain:

$$\dot{\underline{\xi}}_e = (\underline{\rho}_e^{-1}(t))' \underline{\varepsilon} + \underline{\rho}_e^{-1}(t) \dot{\underline{\varepsilon}} = -\underline{\rho}_e^{-2}(t) \dot{\rho}_e(t) \underline{\varepsilon} + \underline{\rho}_e^{-1}(t) \dot{\underline{\varepsilon}} \Rightarrow \dot{\underline{\xi}}_e = \underline{\rho}_e^{-1}(t) [\dot{\underline{\varepsilon}} - \dot{\rho}_e(t) \underline{\xi}_e] \quad (3.72)$$

Substituting equations(3.71),(3.72),(3.68) we obtain:

$$\dot{V} = \underline{S}_e^T \underline{K}^{-1} \cdot \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_e^{-1}(t) [\dot{\underline{\varepsilon}} - \dot{\rho}_e(t) \underline{\xi}_e]$$

Using the equation (3.56) the above equation becomes:

$$\dot{V} = \underline{S}_e^T \underline{K}^{-1} \cdot \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_e^{-1}(t) [-M^{-1}(C\underline{x}_2 + G - \underline{u}) - \ddot{\underline{x}}_d + \underline{\lambda} \underline{e}_2 - \dot{\rho}_e(t) \underline{\xi}_e] \quad (3.73)$$

Exploiting the fact that $\underline{x}_d(t) \in \Omega_x, \underline{e}(t) \in \Omega_e, \forall t \in [0, \tau_{\max})$ and the continuity of $f(\cdot)$ we conclude by the extreme value theorem the boundness of $f(\cdot)$ for all $t \in [0, \tau_{\max})$. Moreover $\underline{\rho}_e(t), \dot{\rho}_e(t)$ and $\ddot{\underline{x}}_d(t)$ are bounded by construction. So, there is a positive constant $\bar{F}$ such that:

$$\| -M^{-1}(C\underline{x}_2 + G) - \ddot{\underline{x}}_d + \underline{\lambda} \underline{e}_2 - \dot{\rho}_e \underline{\xi}_e \| < \bar{F} \quad (3.74), \text{ for all } t \in [0, \tau_{\max})$$

Owing to (3.74) equation (3.73) is transformed into:

$$\dot{V} \leq \underline{S}_e^T \underline{K}^{-1} \cdot \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_e^{-1}(t) [\bar{F} + M^{-1} \underline{u}] \Rightarrow \dot{V} \leq \left(\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_e^{-1}(t) \underline{S}_e \underline{K}^{-1} \right)^T [\bar{F} + M^{-1} \underline{u}] \quad (3.75)$$

By putting $\underline{u} = -\underline{K} \cdot \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_e^{-1}(t) \underline{S}_e$ (3.76), then equation (3.75) becomes:

$$\dot{V} \leq \left(\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_e^{-1}(t) \underline{S}_e \underline{K}^{-1} \right)^T \left[\bar{F} - M^{-1} \underline{K} \cdot \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_e^{-1}(t) \underline{S}_e \right] \quad (3.77)$$

Where $\underline{u}$ the control signal and $\underline{K}$ the control gain matrix. Next, (3.77) is transformed into:

$$\begin{aligned}
\dot{V} &\leq \left(\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \underline{K}^{-1} \right)^T \left[\bar{F} - M^{-1} \underline{K} \cdot \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right] \Rightarrow \\
\Rightarrow \dot{V} &\leq \left(\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \underline{K}^{-1} \right)^T M^{-1} \underline{K} \left(\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right) + \left(\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \underline{K}^{-1} \right)^T \bar{F}
\end{aligned} \tag{3.78}$$

The following properties have been proven in linear algebra:

$$\|a\|^2 \lambda_{\max}(A) \geq a^T A a \geq \|a\|^2 \lambda_{\min}(A) \quad (3.79) \text{ for } A \text{ symmetric matrix}$$

$$\lambda_{\min}(M^{-1}) = \lambda_{\max}(M) \quad (3.80)$$

$$\lambda_{\max}(M^{-1}) = \lambda_{\min}(M) \quad (3.81)$$

Where, a, A, M matrices and $\lambda_{\min}, \lambda_{\max}$ the minimum and maximum eigenvalues. Owing to the property (3.79) and M symmetric matrix the equation (3.78) becomes:

$$\dot{V} \leq \left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\|^2 \lambda_{\max}(M^{-1}) + \left(\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \underline{K}^{-1} \right)^T \bar{F}$$

Using equation (3.81) the above equation becomes:

$$\dot{V} \leq -\lambda_{\min}(M) \left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\|^2 + \left(\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \underline{K}^{-1} \right)^T \bar{F} \quad (3.82)$$

Owing to (3.65) it holds that: $\frac{1}{1-\xi_{\varepsilon}^2} > 1$, so the quantities $\frac{2}{1-\xi_{\varepsilon_i}^2} > 0$ are positive and the matrix

$\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\}$ is positive definite. Furthermore $\rho_{\varepsilon_i}(t) > \lim_{t \rightarrow \infty} \rho_{\varepsilon_i}(t) > 0$ by construction, so $\underline{\rho}_{\varepsilon}^{-1}$ is a positive definite matrix. Therefore the above equation becomes:

$$(3.82) \Rightarrow \dot{V} \leq -\lambda_{\min}(M) \left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\|^2 + \left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \underline{K}^{-1} \bar{F} \right\| \quad (3.83)$$

Also it holds that $\|\underline{S}_{\varepsilon}\| > S_{\varepsilon}$ so the above equation (3.83) becomes:

$$\begin{aligned}
\dot{V} &\leq -\lambda_{\min}(M) \left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\|^2 + \left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \cdot \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\| \underline{K}^{-1} \bar{F} \Rightarrow \\
\Rightarrow \dot{V} &\leq \left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\| \left(-\lambda_{\min}(M) \left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\| + \underline{K}^{-1} \bar{F} \right) \quad (3.84)
\end{aligned}$$

Therefore $\dot{V} < 0$ when:

$$-\lambda_{\min}(M) \left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\| + \underline{K}^{-1} \bar{F} < 0 \quad (3.85)$$

As $-\lambda_{\min}(M), k, \bar{F} > 0$ and also from the assumption 1 and 2 in which $g(x) \geq g^*$, $\forall t \in [0, \tau_{\max})$ so as $g(x) = M^{-1}$ then M positive definite with positive eigenvalues. So $\dot{V} < 0$ when:

$$\left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\| > \frac{\underline{K}^{-1} \bar{F}}{\lambda_{\min}(M)} < 0 \quad (3.86)$$

At next step we define the **bounds of the controller signal**. As the signals in the vector $\underline{S}_{\varepsilon}$ are bounded from its definitions, its upper bounds are given from the equation $|\underline{S}_{\varepsilon}| \leq \bar{\underline{S}}_{\varepsilon}$ (3.87) and $S_{\max} = \max(\bar{\underline{S}}_{\varepsilon}) = \max(\underline{S}_{\varepsilon}(0))$ (3.88). Where $\bar{\underline{S}}$ is a vector with all upper bounds of each signal of vector $\underline{S}_{\varepsilon}$ defined in (3.66). Consequently:

$$\left\| \text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\} \underline{\rho}_{\varepsilon}^{-1}(t) \underline{S}_{\varepsilon} \right\| \leq |\bar{Q}_e \bar{V}_e S_{\max}| = \max \left\{ |\bar{Q}_e \bar{V}_e S_{\max}|, \frac{K_1 \max \bar{F}}{\lambda_{\min}(M)} \right\}, \quad \forall t \in [0, \tau_{\max}) \quad (3.89)$$

Where $\bar{Q}_e = \max(\text{diag} \left\{ \frac{2}{1-\xi_{\varepsilon_i}^2} \right\})$ (3.90) and $\bar{V}_e = \max(\underline{\rho}_{\varepsilon}^{-1}(0))$ (3.91) and $K_1 \max = \max(K^{-1})$ (3.92).

From (3.51) taking the logarithmic function it becomes for each element of $\xi_{\varepsilon}(t)$:

$$-1 < \frac{e^{-S_{\varepsilon_i}} - 1}{e^{-S_{\varepsilon_i}} + 1} = \xi_{\varepsilon_{-l}} \leq \xi_{\varepsilon_i}(t) \leq \xi_{\varepsilon_{-u}} = \frac{e^{S_{\varepsilon_i}} - 1}{e^{S_{\varepsilon_i}} + 1} < 1 \quad (3.93) \text{ for all } t \in [0, \tau_{\max}).$$

So, the control input signals remain bounded $|\underline{u}(t)| \leq \bar{u} = -K_{\max} \cdot Q_e \cdot V_e \cdot S_{\max}$ (3.94) for all $t \in [0, \tau_{\max})$. Since $e(t) \in \Omega_x, \forall t \in [0, \tau_{\max})$ and $\underline{x}_d(t) \in \Omega_x, \forall t \in [0, \tau_{\max})$ the boundness of $\underline{x}(t)$ can be easily deduced for all $t \in [0, \tau_{\max})$. Where $K_{\max} = \max(\underline{K})$ (3.95).

Next step is to be shown that $\tau_{\max}$ can be extended to ∞ . From equation (3.93) it is derived that $\xi_{\varepsilon_i}(t) \in \Omega'_{\varepsilon}, i=1..n, \forall t \in [0, \tau_{\max})$ where the set $\Omega'_{\varepsilon} = \Omega_{\varepsilon} \times \left[\frac{e^{-S_{\varepsilon_i}} - 1}{e^{-S_{\varepsilon_i}} + 1}, \frac{e^{S_{\varepsilon_i}} - 1}{e^{S_{\varepsilon_i}} + 1} \right]$ is nonempty and compact. Also, it can be easily verified that $\Omega'_{\varepsilon} \subset \Omega_{\varepsilon}$. Hence, assuming $\tau_{\max} < \infty$ and since $\Omega'_{\varepsilon} \subset \Omega_{\varepsilon}$ Proposition 1 in subsection 2-C dictates the existence of a time instant $t' \in [0, \tau_{\max})$ such that $\xi(t') \notin \Omega'_{\varepsilon}$ which is a clear contradiction. Therefore, $\tau_{\max} = \infty$. Hence, all closed loop signals remain bounded and moreover $\xi(t) \in \Omega'_{\varepsilon} \subset \Omega_{\varepsilon}, \forall t \geq 0$. From equations (3.57) and (3.93) it is concluded:

$$\begin{aligned}
(3.49) \Rightarrow & \left\{ \begin{array}{c} -1 \cdot \underline{\rho}_\varepsilon(t) < \frac{e^{-s_{\varepsilon_1}} - 1}{e^{-s_{\varepsilon_1}} + 1} \cdot \underline{\rho}_\varepsilon(t) \leq \xi_{\varepsilon_1}(t) \cdot \underline{\rho}_\varepsilon(t) \leq \frac{e^{s_{\varepsilon_1}} - 1}{e^{s_{\varepsilon_1}} + 1} \cdot \underline{\rho}_\varepsilon(t) < 1 \cdot \underline{\rho}_\varepsilon(t) \\ \vdots \\ -1 \cdot \underline{\rho}_\varepsilon(t) < \frac{e^{-s_{\varepsilon_n}} - 1}{e^{-s_{\varepsilon_n}} + 1} \cdot \underline{\rho}_\varepsilon(t) \leq \xi_{\varepsilon_n}(t) \cdot \underline{\rho}_\varepsilon(t) \leq \frac{e^{s_{\varepsilon_n}} - 1}{e^{s_{\varepsilon_n}} + 1} \cdot \underline{\rho}_\varepsilon(t) < 1 \cdot \underline{\rho}_\varepsilon(t) \end{array} \right\} \Rightarrow \\
& \Rightarrow -\underline{\rho}_\varepsilon(t) < \frac{e^{-s_{\varepsilon_i}} - 1}{e^{-s_{\varepsilon_i}} + 1} \cdot \underline{\rho}_\varepsilon(t) \leq \xi_{\varepsilon_i}(t) \cdot \underline{\rho}_\varepsilon(t) \leq \frac{e^{s_{\varepsilon_i}} - 1}{e^{s_{\varepsilon_i}} + 1} \cdot \underline{\rho}_\varepsilon(t) < \underline{\rho}_\varepsilon(t) \quad (3.96)
\end{aligned}$$

For all $t \geq 0$ and consequently that $\underline{\varepsilon}(\underline{e}(t))$ converges exponentially (owing to $\underline{\rho}_\varepsilon(t)$ which has been chosen as a matrix of exponentially decreasing performance functions) to the compact set:

$$E = \left\{ \underline{\varepsilon} \triangleq [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]^T \in \mathfrak{R}^n : |\varepsilon_i| \leq \lim_{t \rightarrow \infty} \rho_{\varepsilon_j}(t) \right\}, j=1, \dots, n$$

Moreover owing to equation (3.41) being bounded input bounded output, it is obtained (using the analysis in pages 276-280 of [47]) that the state errors $\underline{e}_1, \underline{e}_2$ (as defined in (3.38) and (3.39) equations) converge to the compact sets:

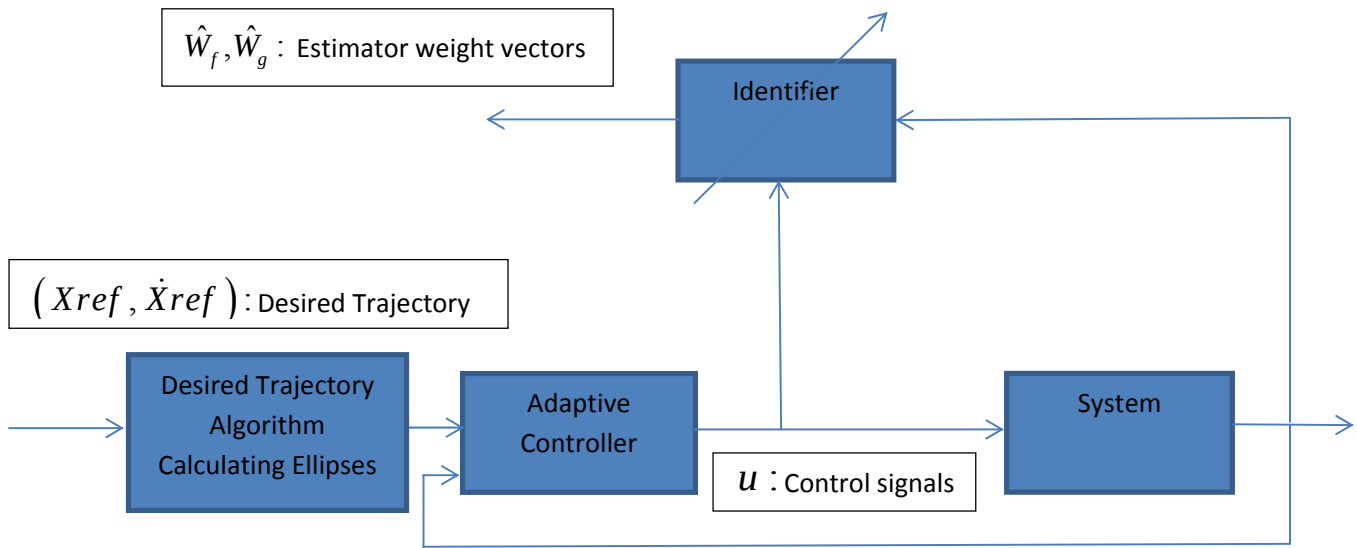
$$E_1 = \left\{ [e_1^1, e_1^2, \dots, e_1^n]^T \in \mathfrak{R}^n : |e_1^j| \leq \frac{\lim_{t \rightarrow \infty} \rho_{\varepsilon_j}(t)}{\lambda_j} \right\}, \quad E_2 = \left\{ [e_2^1, e_2^2, \dots, e_2^n]^T \in \mathfrak{R}^n : |e_2^j| \leq 2 \lim_{t \rightarrow \infty} \rho_{\varepsilon_j}(t) \right\} \text{ for } j=1, \dots, n$$

where n the degrees of freedom of the Lagrange system

The above proposed control scheme achieves its goals without residing to the need of rendering $\bar{\underline{S}}_\varepsilon$ (as defined in (3.87)) arbitrarily small, through extreme values of the control gains $k_1, k_2, \dots, k_n$ in the matrix $\underline{K}$. The unknown system nonlinearities $f(\cdot), g(\cdot)$ affect only the size of $\bar{\underline{S}}_\varepsilon$ but leave unaltered the achieved convergence properties. The actual tracking performance at the steady state is assessed by the sets E_1, E_2 defined above, which are closely related and depending on the matrix of performance functions $\underline{\rho}_\varepsilon(t)$ defined in (3.44) as well as the control gain matrix $\underline{\lambda}$ defined in (3.42). So it is concluded that any prescribed performance can be achieved by selecting appropriately the parameters $\rho_\varepsilon(t)$ and λ .

C. Identification scheme design

In this part, the controller which was designed previously is used to ensure that system states follow the desired trajectory and it is applied in cooperation with RBF neural networks to achieve the purpose of local learning of nonlinear functions $f(\cdot), g(\cdot)$ in a predefined compact set. The cooperation between the controller and the identifier is appeared in the following picture where the connections of the signals between controller, RBF Networks (Identifier) and System are presented and X_{ref} the reference signal and u the control input.



Picture 4. Block diagram of identification scheme

First step in the process of learning is to create an expression using RBF neural networks in order to express each unknown function in a linear form with RBF neural weights and regressor vectors. The following equations are used for the identification of the unknown functions:

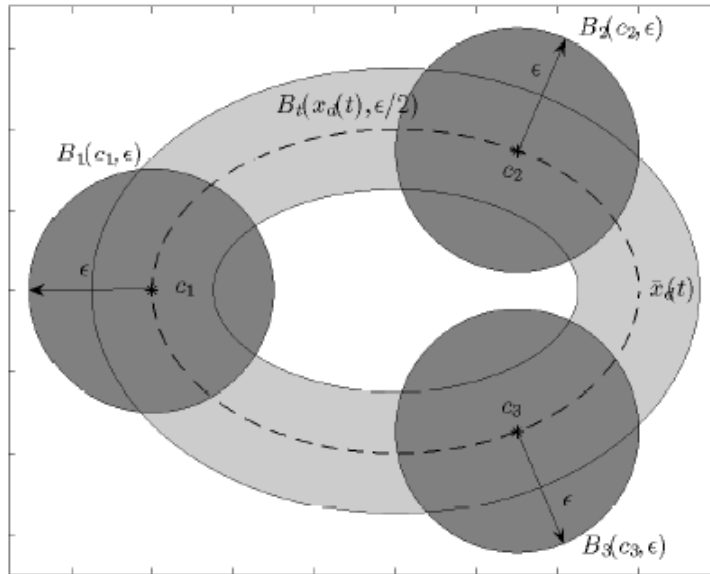
$$f(x) = W_f^{*T} Z_f(x) + \varepsilon_f(x) \quad (3.97)$$

$$g(x) = W_g^{*T} Z_g(x) + \varepsilon_g(x) \quad (3.98)$$

Where, $Z_f(x), Z_g(x)$ are the regressor vectors of selected RBFs with centers $c_i \in \Omega_x, i=1, \dots, n_c$,

and take the form:
$$Z_f(x) = \begin{bmatrix} Z_{f_1}(x) \\ \vdots \\ Z_{f_{n_c}}(x) \end{bmatrix}, \quad Z_g(x) = \begin{bmatrix} Z_{g_1}(x) & Z_{g_1}(x) \\ \vdots & \vdots \\ Z_{g_1}(x) & Z_{g_1}(x) \end{bmatrix}$$

W_f^{*T}, W_g^{*T} are the optimal values of RBF weights which minimize the error terms $\bar{\varepsilon}_f = \sup_{x \in \Omega_x} \{\varepsilon_f(x)\}, \bar{\varepsilon}_g = \sup_{x \in \Omega_x} \{\varepsilon_g(x)\}$. The purpose of the learning process is focused on the estimation of the unknown RBF weight vectors W_f, W_g in an attempt to converge to the optimal values. In the following proposition, it is shown how we should design the desired trajectory $\underline{x}_d(t)$ and select the controller parameters as the performance function $\rho_\varepsilon(t)$ and the control gain matrix $\underline{\lambda}$ in a way that the control scheme (presented in the previous section) guarantees that the regressor vectors $Z_f(x(t)), Z_g(x(t))$ are persistently exciting in the sense of condition (2.74).



Picture 5. Illustration of the key idea behind the proof of the persistency of excitation of $Z_f(x(t))$ and $Z_g(x(t))$. The state $x(t)$ will remain inside the non-intersecting sets $B_i(c_i, \epsilon), i=1, 2, 3$ for a time interval of magnitude

Proposition 2: Select a periodic, bounded and smooth trajectory $x_{ref}(t):[0,\infty)\rightarrow\Omega_x$ traversing all the centers of the selected RBFs, i.e., there exist time instants $t_{c_i}+mT$ such that $x_{ref}(t_{c_i}+mT)=c_i$ for all $i=1,\dots,n_c$ with $m\in\mathbb{N}$ and T denoting the period of the orbit $x_{ref}(t)$. By designing the performance function matrix $\rho_\varepsilon(t)$ and the control gain matrix $\underline{\lambda}$ such that $\lim_{t\rightarrow\infty}\rho_\varepsilon(t)<\frac{\epsilon}{2\sqrt{\sum_{i=1}^n\left(\frac{2^{i-1}}{\lambda^{n-1}}\right)^2}}$ with ϵ satisfying $\epsilon<\min_{i\neq j}\left\{\frac{1}{2}\|c_i-c_j\|\right\}, i,j=1,\dots,n_c$ then the control law

(3.46) guarantees that the regressor vectors $Z_f(x), Z_g(x)$ are persistently exciting according to the equation (2.74).

Proof: Consider any $\epsilon<\min_{i\neq j}\left\{\frac{1}{2}\|c_i-c_j\|\right\}, i,j=1,\dots,n_c$ and define the ball area $B_t\left(x_{ref}(t),\frac{\epsilon}{2}\right)=\left\{x\in\mathbb{R}^n:\|x-x_{ref}(t)\|\leq\frac{\epsilon}{2}\right\}$ (See picture 5). According to *Theorem 3* if we select the following conditions for $\rho_{\varepsilon_j}(t)$, $\lim_{t\rightarrow\infty}\rho_{\varepsilon_j}(t)<\frac{\epsilon}{2\sqrt{\sum_{i=1}^2\left(\frac{2^{i-1}}{\lambda_j^{n-1}}\right)^2}}, j=1,\dots,N$, (where N the degrees of

freedom) it can be easily verified that there exists a time instant T_0 such that $x(t)\in B_t\left(x_{ref}(t),\frac{\epsilon}{2}\right), \forall t\geq T_0$. Now let us consider the ϵ -neighborhoods of the RBF centers $B_i(c_i,\epsilon)=\{x\in\mathbb{R}^n:\|x-c_i\|\leq\epsilon\}, i=1,\dots,n_c$ which are nonintersecting. Owing to the smoothness and

the boundness of $x_{ref}(t), \dot{x}_{ref}(t)$ in the vectors $\underline{x}_d=\begin{bmatrix} x_{ref}(t) \\ \vdots \\ x_{ref}(t) \end{bmatrix}$, $\underline{\dot{x}}_d=\begin{bmatrix} \dot{x}_{ref}(t) \\ \vdots \\ \dot{x}_{ref}(t) \end{bmatrix}$ as well as that

$x_{ref}(t_{c_i}+nT)=c_i$ for all $i=1,\dots,n_c$ with $m\in\mathbb{N}$ and T denoting the period the period of the orbit $x_{ref}(t)$, there exists a positive constant δT such that $B_t\left(x_{ref}(t),\frac{\epsilon}{2}\right)\subset B_i(c_i,\epsilon), \forall t\in\left[t_{c_i}+mT-\frac{\delta T}{2}, t_{c_i}+mT+\frac{\delta T}{2}\right]$ and for all $i=1,\dots,n_c$ with $m\in\mathbb{N}$.

Consequently since $x(t)\in B_t\left(x_{ref}(t),\frac{\epsilon}{2}\right), \forall t\geq T_0$ it is concluded that

$x(t) \in B_i(c_i, \epsilon), \forall t \in \left[t_{c_i} + mT - \frac{\delta T}{2}, t_{c_i} + mT + \frac{\delta T}{2} \right] \cap [T_0, \infty)$ and for all $i=1, \dots, n_c$ with $m \in \mathbb{N}$. Hence, applying Theorem 1 the proof is completed.

At next step, we define the error dynamics of the examined system. Having defined the state error vectors $\underline{e} = \begin{bmatrix} \underline{e}_1 \\ \underline{e}_2 \end{bmatrix}$ where $\underline{e}_1 = \underline{x}_1 - \underline{x}_d$ and $\underline{e}_2 = \underline{x}_2 - \dot{\underline{x}}_d$ the error dynamics are:

$$\dot{\underline{e}}_1 = \dot{\underline{x}}_1 - \dot{\underline{x}}_d$$

$$\dot{\underline{e}}_2 = \dot{\underline{x}}_2 - \ddot{\underline{x}}_d$$

Using equation (2.62), the above equations become:

$$\dot{\underline{e}}_1 = \dot{\underline{x}}_1 - \dot{\underline{x}}_d \Rightarrow \dot{\underline{e}}_1 = \underline{x}_2 - \dot{\underline{x}}_d = \underline{e}_2 \quad (3.99)$$

$$\dot{\underline{e}}_2 = \dot{\underline{x}}_2 - \ddot{\underline{x}}_d \Rightarrow \dot{\underline{e}}_2 = \underline{f}(\underline{x}) + \underline{g}(\underline{x}) \cdot \underline{u} - \ddot{\underline{x}}_d \quad (3.100)$$

Writing the above equations (3.99), (3.100) in matrix form it is derived that:

$$\begin{bmatrix} \dot{\underline{e}}_1 \\ \dot{\underline{e}}_2 \end{bmatrix} = A \begin{bmatrix} \underline{e}_1 \\ \underline{e}_2 \end{bmatrix} + B \left[\underline{f}(\underline{x}) + \underline{g}(\underline{x}) \cdot \underline{u} - \ddot{\underline{x}}_d \right] \quad (3.101)$$

Matrices A, B are in the form:

$$A = \begin{bmatrix} 0 & . & . & . & 0 & 1 & 0 & . & . & 0 \\ . & . & . & . & . & 0 & 1 & 0 & . & . \\ . & . & . & . & . & . & 0 & 1 & 0 & . \\ . & . & . & . & . & . & . & 0 & 1 & 0 \\ . & . & . & . & . & . & . & . & 0 & 1 \\ . & . & . & . & . & . & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & 0 & . & . & . & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & . & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & . & . \\ 1 & 0 & . & . & . \\ 0 & 1 & 0 & . & . \\ . & 0 & 1 & 0 & . \\ . & . & 0 & 1 & 0 \\ 0 & . & . & 0 & 1 \end{bmatrix}$$

The identification scheme is defined through the following equation:

$$\dot{\underline{\hat{x}}} = A\hat{x} + B \left(\hat{\underline{W}}_f^T \underline{Z}_f(x) + \hat{\underline{W}}_g^T \underline{Z}_g(x) \underline{u} + K^T \underline{\hat{x}} \right) \quad (3.102)$$

Where, $\tilde{x} = x - \hat{x}$ the state estimation error vector, matrices A, B defined above, K the gain vector selected such that $A - BK^T$ is Hurwitz and $\hat{W}_f, \hat{W}_g$ are estimates of the unknown optimal weight vectors W_f^*, W_g^* and using the Lyapunov analysis method it is derived (analyzed in the following proof) that:

$$\dot{\hat{W}}_f = \gamma_f \left[Z_f(x) \tilde{x}^T B - \sigma_f \hat{W}_f \right] \quad (3.103)$$

$$\dot{\hat{W}}_g = \gamma_g \left[Z_g(x) \tilde{x}^T B - \sigma_g \hat{W}_g \right] \quad (3.104)$$

Where, $\gamma_f, \gamma_g, \sigma_f, \sigma_g$ positive gains.

Theorem 4: Consider system (2.62) excited by the control input (3.46). The identifier (3.102)-(3.104) guarantees that: i) all signals in the identification loop are bounded, ii) the estimates $\hat{W}_f, \hat{W}_g$ converge to small neighborhoods of their optimal values W_f^*, W_g^* and iii) the state estimation error $\tilde{x}$ converges to a small neighborhood of zero.

Proof: Firstly consider the Lyapunov function candidate in the form:

$$L = \frac{1}{2} \tilde{x}^T \tilde{x} + \frac{1}{2\gamma_f} \text{trace}(\tilde{W}_f^T \tilde{W}_f) + \frac{1}{2\gamma_g} \text{trace}(\tilde{W}_g^T \tilde{W}_g) \quad (3.105)$$

Where, trace quantity of an orthogonal matrix is the sum of elements in the main diagonal. Also $\tilde{W}_f = W_f^* - \hat{W}_f$, $\tilde{W}_g = W_g^* - \hat{W}_g$ are the parametric errors.

Differentiating the above equation (3.105) we obtain:

$$\begin{aligned} \dot{L} &= \tilde{x}^T \dot{\tilde{x}} + \frac{1}{2\gamma_f} \left[\text{trace}(\dot{\tilde{W}}_f^T \tilde{W}_f) + \text{trace}(\tilde{W}_f^T \dot{\tilde{W}}_f) \right] + \frac{1}{2\gamma_g} \left[\text{trace}(\dot{\tilde{W}}_g^T \tilde{W}_g) + \text{trace}(\tilde{W}_g^T \dot{\tilde{W}}_g) \right] \xrightarrow{\tilde{x} = x - \hat{x} \Rightarrow \dot{\tilde{x}} = \dot{x} - \dot{\hat{x}}} \\ &\Rightarrow \dot{L} = \tilde{x}^T (\dot{x} - \dot{\hat{x}}) + \frac{1}{2\gamma_f} \left[\text{trace}(\dot{\tilde{W}}_f^T \tilde{W}_f) + \text{trace}(\tilde{W}_f^T \dot{\tilde{W}}_f) \right] + \frac{1}{2\gamma_g} \left[\text{trace}(\dot{\tilde{W}}_g^T \tilde{W}_g) + \text{trace}(\tilde{W}_g^T \dot{\tilde{W}}_g) \right] \end{aligned}$$

Using (3.102) the above equation becomes:

$$\begin{aligned} \dot{L} &= \tilde{x}^T \left[A\tilde{x} + B\tilde{f}(x) + B\tilde{g}(x)\underline{u} - A\hat{x} - B\hat{W}_f^T Z_f(x) - B\hat{W}_g^T Z_g(x)\underline{u} - BK^T \tilde{x} \right] + \\ &\frac{1}{2\gamma_f} \left[\text{trace}(\dot{\tilde{W}}_f^T \tilde{W}_f) + \text{trace}(\tilde{W}_f^T \dot{\tilde{W}}_f) \right] + \frac{1}{2\gamma_g} \left[\text{trace}(\dot{\tilde{W}}_g^T \tilde{W}_g) + \text{trace}(\tilde{W}_g^T \dot{\tilde{W}}_g) \right] \quad (3.106) \end{aligned}$$

Using equations (3.97), (3.98) the above equation (3.106) becomes:

$$\begin{aligned}
\dot{L} &= \tilde{x}^T \left[A\underline{x} + B\underline{W}_f^{*T} \underline{Z}_f(x) + B\underline{\mathcal{E}}_{f(x)} + B\underline{W}_g^{*T} \underline{Z}_g(x) \underline{u} + B\underline{\mathcal{E}}_{g(x)} \underline{u} - A\hat{x} - B\hat{W}_f^T \underline{Z}_f(x) - B\hat{W}_g^T \underline{Z}_g(x) \underline{u} - BK^T \tilde{x} \right] + \\
&\frac{1}{2\gamma_f} \left[\text{trace}(\dot{\tilde{W}}_f^T \tilde{W}_f) + \text{trace}(\tilde{W}_f^T \dot{\tilde{W}}_f) \right] + \frac{1}{2\gamma_g} \left[\text{trace}(\dot{\tilde{W}}_g^T \tilde{W}_g) + \text{trace}(\tilde{W}_g^T \dot{\tilde{W}}_g) \right] \Rightarrow \\
&\Rightarrow \dot{L} = \tilde{x}^T \left[A(\underline{x} - \hat{x}) + B(\underline{W}_f^{*T} - \hat{W}_f^T) \underline{Z}_f(x) + B(\underline{W}_g^{*T} - \hat{W}_g^T) \underline{Z}_g(x) \underline{u} + B\underline{\mathcal{E}}_{f(x)} + B\underline{\mathcal{E}}_{g(x)} \underline{u} - BK^T \tilde{x} \right] + \\
&+ \frac{1}{\gamma_f} \text{trace}(\tilde{W}_f^T \dot{\tilde{W}}_f) + \frac{1}{\gamma_g} \text{trace}(\tilde{W}_g^T \dot{\tilde{W}}_g) \stackrel{\tilde{x} = \underline{x} - \hat{x}}{\Rightarrow} \\
&\Rightarrow \dot{L} = \tilde{x}^T \left[(A - BK^T) \tilde{x} + B(\underline{W}_f^{*T} - \hat{W}_f^T) \underline{Z}_f(x) + B(\underline{W}_g^{*T} - \hat{W}_g^T) \underline{Z}_g(x) \underline{u} + B\underline{\mathcal{E}}_{f(x)} + B\underline{\mathcal{E}}_{g(x)} \underline{u} \right] + \\
&+ \frac{1}{\gamma_f} \text{trace}(\tilde{W}_f^T \dot{\tilde{W}}_f) + \frac{1}{\gamma_g} \text{trace}(\tilde{W}_g^T \dot{\tilde{W}}_g) \Rightarrow \\
&\Rightarrow \dot{L} = \tilde{x}^T (A - BK^T) \tilde{x} + \tilde{x}^T B(\underline{W}_f^{*T} - \hat{W}_f^T) \underline{Z}_f(x) + \tilde{x}^T B(\underline{W}_g^{*T} - \hat{W}_g^T) \underline{Z}_g(x) \underline{u} + \tilde{x}^T B\underline{\mathcal{E}}_{f(x)} + \tilde{x}^T B\underline{\mathcal{E}}_{g(x)} \underline{u} \\
&+ \frac{1}{\gamma_f} \text{trace}(\tilde{W}_f^T \dot{\tilde{W}}_f) + \frac{1}{\gamma_g} \text{trace}(\tilde{W}_g^T \dot{\tilde{W}}_g) \tag{3.107}
\end{aligned}$$

It holds that:

$$\tilde{W}_f = \underline{W}_f^* - \hat{W}_f \Rightarrow \dot{\tilde{W}}_f = -\dot{\hat{W}}_f \tag{3.108}$$

$$\tilde{W}_g = \underline{W}_g^* - \hat{W}_g \Rightarrow \dot{\tilde{W}}_g = -\dot{\hat{W}}_g \tag{3.109}$$

Substituting equations (3.108)-(3.109) into equation (3.107) we obtain:

$$\begin{aligned}
\dot{L} &= \tilde{x}^T (A - BK^T) \tilde{x} + \tilde{x}^T B\underline{W}_f^{*T} \underline{Z}_f(x) + \tilde{x}^T B\underline{W}_g^{*T} \underline{Z}_g(x) \underline{u} + \tilde{x}^T B\underline{\mathcal{E}}_{f(x)} + \tilde{x}^T B\underline{\mathcal{E}}_{g(x)} \underline{u} \\
&- \frac{1}{\gamma_f} \text{trace}(\tilde{W}_f^T \dot{\tilde{W}}_f) - \frac{1}{\gamma_g} \text{trace}(\tilde{W}_g^T \dot{\tilde{W}}_g) \tag{3.110}
\end{aligned}$$

Also the following *trace* property is valid and it is applied into equation (3.110) which becomes:

$$a^T b = \text{trace}(ba^T) \tag{3.111}$$

$$\begin{aligned}
\dot{L} &= \tilde{x}^T (A - BK^T) \tilde{x} + \text{trace}(\tilde{W}_f^T \underline{Z}_f(x) \tilde{x}^T B) + \text{trace}(\tilde{W}_g^T \underline{Z}_g(x) \underline{u} \tilde{x}^T B) + \tilde{x}^T B\underline{\mathcal{E}}_{f(x)} + \tilde{x}^T B\underline{\mathcal{E}}_{g(x)} \underline{u} \\
&- \frac{1}{\gamma_f} \text{trace}(\tilde{W}_f^T \dot{\tilde{W}}_f) - \frac{1}{\gamma_g} \text{trace}(\tilde{W}_g^T \dot{\tilde{W}}_g) \Rightarrow
\end{aligned}$$

$$\Rightarrow \dot{L} = \tilde{x}^T (A - BK^T) \tilde{x} + \text{trace} \left[\tilde{W}_f^T \left(Z_f(x) \tilde{x}^T B - \frac{1}{\gamma_f} \dot{\hat{W}}_f \right) \right] + \text{trace} \left[\tilde{W}_g^T \left(Z_g(x) \underline{u} \tilde{x}^T B - \frac{1}{\gamma_g} \dot{\hat{W}}_g \right) \right] + (3.112)$$

$$+ \tilde{x}^T B \underline{\varepsilon}_{f(x)} + \tilde{x}^T B \underline{\varepsilon}_{g(x)} \underline{u}$$

Assuming that:

$$Z_f(x) \tilde{x}^T B - \frac{1}{\gamma_f} \dot{\hat{W}}_f = \sigma_f \hat{W}_f \Rightarrow \dot{\hat{W}}_f = \gamma_f (Z_f(x) \tilde{x}^T B - \sigma_f \hat{W}_f) \quad (3.113)$$

$$Z_g(x) \underline{u} \tilde{x}^T B - \frac{1}{\gamma_g} \dot{\hat{W}}_g = \sigma_g \hat{W}_g \Rightarrow \dot{\hat{W}}_g = \gamma_g (Z_g(x) \underline{u} \tilde{x}^T B - \sigma_g \hat{W}_g) \quad (3.114)$$

Then, equation (3.112) becomes:

$$\Rightarrow \dot{L} = \tilde{x}^T (A - BK^T) \tilde{x} + \sigma_f \text{trace}(\tilde{W}_f^T \hat{W}_f) + \sigma_g \text{trace}(\tilde{W}_g^T \hat{W}_g) + \tilde{x}^T B \underline{\varepsilon}_{f(x)} + \tilde{x}^T B \underline{\varepsilon}_{g(x)} \underline{u} \quad (3.115)$$

Using the equations (3.108) and (3.109) the above equation (3.115) becomes:

$$\begin{aligned} \dot{L} &= \tilde{x}^T (A - BK^T) \tilde{x} + \sigma_f \text{trace}(\tilde{W}_f^T (W_f^* - \tilde{W}_f)) + \sigma_g \text{trace}(\tilde{W}_g^T (W_g^* - \tilde{W}_g)) + \tilde{x}^T B \underline{\varepsilon}_{f(x)} + \tilde{x}^T B \underline{\varepsilon}_{g(x)} \underline{u} \Rightarrow \\ &\Rightarrow \dot{L} = \tilde{x}^T (A - BK^T) \tilde{x} - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + \sigma_f \text{trace}(\tilde{W}_f^T W_f^*) + \sigma_g \text{trace}(\tilde{W}_g^T W_g^*) + \\ &+ \tilde{x}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u}) \end{aligned} \quad (3.116)$$

The following property is valid for trace quantity:

$$\text{trace}(Q^T Q) \geq 0 \quad (3.117)$$

So in equation (3.116) the quantity is negative, $-\sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) < 0$. Assuming

that $\tilde{W}_f^T$ is in the form: $\tilde{W}_f^T = \begin{bmatrix} \tilde{W}_{f11} & \tilde{W}_{f21} & \cdot & \cdot & \cdot & \tilde{W}_{f_{n_c1}} \\ \tilde{W}_{f12} & \tilde{W}_{f22} & \cdot & \cdot & \cdot & \tilde{W}_{f_{n_c2}} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \tilde{W}_{f1n} & \tilde{W}_{f2n} & \cdot & \cdot & \cdot & \tilde{W}_{f_{n_cn}} \end{bmatrix} \quad (3.118)$, and also the W_f^* in the form:

$$W_f^* = \begin{bmatrix} W_{f_{11}}^* & W_{f_{12}}^* & \cdot & \cdot & \cdot & W_{f_{1n}}^* \\ W_{f_{21}}^* & W_{f_{22}}^* & \cdot & \cdot & \cdot & W_{f_{2n}}^* \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ W_{f_{n_c1}}^* & W_{f_{n_c2}}^* & \cdot & \cdot & \cdot & W_{f_{n_cn}}^* \end{bmatrix} \quad (3.119), \text{ where } n \text{ the number of degrees of freedom of the system}$$

and n_c the number of nodes we choose in the lattice of the compact set of learning. Then the quantity $\sigma_f \text{trace}(\tilde{W}_f^T W_f^*)$ becomes:

$$\begin{aligned} \sigma_f \text{trace}(\tilde{W}_f^T W_f^*) &= \tilde{W}_{f_{11}} W_{f_{11}}^* + \tilde{W}_{f_{22}} W_{f_{22}}^* + \dots + \tilde{W}_{f_{n_c1}} W_{f_{n_c1}}^* + \tilde{W}_{f_{12}} W_{f_{12}}^* + \tilde{W}_{f_{22}} W_{f_{22}}^* + \dots + \tilde{W}_{f_{n_c2}} W_{f_{n_c2}}^* + \dots + \\ &+ \tilde{W}_{f_{1n}} W_{f_{1n}}^* + \tilde{W}_{f_{2n}} W_{f_{2n}}^* + \dots + \tilde{W}_{f_{n_cn}} W_{f_{n_cn}}^* \quad (3.120) \end{aligned}$$

Where n_c is the number of the nodes in the compact set we define as the set of learning. In the same way we assume for $\tilde{W}_g^T$ and W_g^* :

$$\tilde{W}_g^T = \begin{bmatrix} \tilde{W}_{g_{11}} & \tilde{W}_{g_{21}} & \cdot & \cdot & \cdot & \tilde{W}_{g_{n_c1}} \\ \tilde{W}_{g_{12}} & \tilde{W}_{g_{22}} & \cdot & \cdot & \cdot & \tilde{W}_{g_{n_c2}} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \tilde{W}_{g_{1n}} & \tilde{W}_{g_{2n}} & \cdot & \cdot & \cdot & \tilde{W}_{g_{n_cn}} \end{bmatrix} \quad (3.121), \quad W_g^* = \begin{bmatrix} W_{g_{11}}^* & W_{g_{12}}^* & \cdot & \cdot & \cdot & W_{g_{1n}}^* \\ W_{g_{21}}^* & W_{g_{22}}^* & \cdot & \cdot & \cdot & W_{g_{2n}}^* \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ W_{g_{n_c1}}^* & W_{g_{n_c2}}^* & \cdot & \cdot & \cdot & W_{g_{n_cn}}^* \end{bmatrix} \quad (3.122)$$

Then the quantity $\sigma_g \text{trace}(\tilde{W}_g^T W_g^*)$ becomes:

$$\begin{aligned} \sigma_g \text{trace}(\tilde{W}_g^T W_g^*) &= \tilde{W}_{g_{11}} W_{g_{11}}^* + \tilde{W}_{g_{21}} W_{g_{21}}^* + \dots + \tilde{W}_{g_{n_c1}} W_{g_{n_c1}}^* + \tilde{W}_{g_{12}} W_{g_{12}}^* + \tilde{W}_{g_{22}} W_{g_{22}}^* + \dots + \tilde{W}_{g_{n_c2}} W_{g_{n_c2}}^* + \dots + \\ &+ \tilde{W}_{g_{1n}} W_{g_{1n}}^* + \tilde{W}_{g_{2n}} W_{g_{2n}}^* + \dots + \tilde{W}_{g_{n_cn}} W_{g_{n_cn}}^* \quad (3.123) \end{aligned}$$

Substituting the equations (3.120) and (3.123) into equation (3.116) we obtain:

$$\begin{aligned} \dot{L} &= \tilde{\underline{x}}^T (A - BK^T) \tilde{\underline{x}} - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + \\ &+ \tilde{W}_{f_{11}} W_{f_{11}}^* + \tilde{W}_{f_{21}} W_{f_{21}}^* + \dots + \tilde{W}_{f_{n_c1}} W_{f_{n_c1}}^* + \tilde{W}_{f_{12}} W_{f_{12}}^* + \tilde{W}_{f_{22}} W_{f_{22}}^* + \dots + \tilde{W}_{f_{n_c2}} W_{f_{n_c2}}^* + \dots + \tilde{W}_{f_{1n}} W_{f_{1n}}^* + \tilde{W}_{f_{2n}} W_{f_{2n}}^* + \dots + \tilde{W}_{f_{n_cn}} W_{f_{n_cn}}^* \\ &+ \tilde{W}_{g_{11}} W_{g_{11}}^* + \tilde{W}_{g_{21}} W_{g_{21}}^* + \dots + \tilde{W}_{g_{n_c1}} W_{g_{n_c1}}^* + \tilde{W}_{g_{12}} W_{g_{12}}^* + \tilde{W}_{g_{22}} W_{g_{22}}^* + \dots + \tilde{W}_{g_{n_c2}} W_{g_{n_c2}}^* + \dots + \tilde{W}_{g_{1n}} W_{g_{1n}}^* + \tilde{W}_{g_{2n}} W_{g_{2n}}^* + \dots + \tilde{W}_{g_{n_cn}} W_{g_{n_cn}}^* \\ &+ \tilde{\underline{x}}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u}) \quad (3.124) \end{aligned}$$

The following inequality is proved that it holds and it is called “Young’s inequality”:

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q} \quad (3.125)$$

Where a, b non-negative real numbers and p, q are positive real numbers such that $\frac{1}{p} + \frac{1}{q} = 1$.

Also for the absolute value of a real number d it holds that:

$$d \leq |d| \quad (3.126)$$

Using the above property (3.126) equation (3.124) becomes:

$$(3.124) \Rightarrow$$

$$\begin{aligned} \dot{L} = & \tilde{x}^T (A - BK^T) \tilde{x} - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + \\ & + |\tilde{W}_{f1} W_{f1}^*| + |\tilde{W}_{f21} W_{f21}^*| + \dots + |\tilde{W}_{f_{n_c1}} W_{f_{n_c1}}^*| + |\tilde{W}_{f12} W_{f12}^*| + |\tilde{W}_{f22} W_{f22}^*| + \dots + |\tilde{W}_{f_{n_c2}} W_{f_{n_c2}}^*| + \dots + \\ & + |\tilde{W}_{f1n} W_{f1n}^*| + |\tilde{W}_{f2n} W_{f2n}^*| + \dots + |\tilde{W}_{f_{n_cn}} W_{f_{n_cn}}^*| + \\ & + |\tilde{W}_{g11} W_{g11}^*| + |\tilde{W}_{g21} W_{g21}^*| + \dots + |\tilde{W}_{g_{n_c1}} W_{g_{n_c1}}^*| + |\tilde{W}_{g12} W_{g12}^*| + |\tilde{W}_{g22} W_{g22}^*| + \dots + |\tilde{W}_{g_{n_c2}} W_{g_{n_c2}}^*| + \dots + \\ & + |\tilde{W}_{g1n} W_{g1n}^*| + |\tilde{W}_{g2n} W_{g2n}^*| + \dots + |\tilde{W}_{g_{n_cn}} W_{g_{n_cn}}^*| + \\ & + \tilde{x}^T B (\mathcal{E}_{f(x)} + \mathcal{E}_{g(x)} u) \end{aligned} \quad (3.127)$$

Applying Young’s inequality (3.125) the above equation (3.127) becomes:

$$(3.127) \Rightarrow$$

$$\begin{aligned}
\dot{L} = & \tilde{x}^T (A - BK^T) \tilde{x} - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + \\
& + \frac{|\tilde{W}_{f11}|^2}{2} + \frac{|W_{f11}^*|^2}{2} + \frac{|\tilde{W}_{f21}|^2}{2} + \frac{|W_{f21}^*|^2}{2} + \dots + \frac{|\tilde{W}_{f_{n_c1}}|^2}{2} + \frac{|W_{f_{n_c1}}^*|^2}{2} + \frac{|\tilde{W}_{f12}|^2}{2} + \frac{|W_{f12}^*|^2}{2} + \frac{|\tilde{W}_{f22}|^2}{2} + \frac{|W_{f22}^*|^2}{2} + \dots + \\
& + \frac{|\tilde{W}_{f_{n_c2}}|^2}{2} + \frac{|W_{f_{n_c2}}^*|^2}{2} + \dots + \frac{|\tilde{W}_{f1n}|^2}{2} + \frac{|W_{f1n}^*|^2}{2} + \frac{|\tilde{W}_{f2n}|^2}{2} + \frac{|W_{f2n}^*|^2}{2} + \dots + \frac{|\tilde{W}_{f_{n_cn}}|^2}{2} + \frac{|W_{f_{n_cn}}^*|^2}{2} + \\
& + \frac{|\tilde{W}_{g11}|^2}{2} + \frac{|W_{g11}^*|^2}{2} + \frac{|\tilde{W}_{g21}|^2}{2} + \frac{|W_{g21}^*|^2}{2} + \dots + \frac{|\tilde{W}_{g_{n_c1}}|^2}{2} + \frac{|W_{g_{n_c1}}^*|^2}{2} + \frac{|\tilde{W}_{g12}|^2}{2} + \frac{|W_{g12}^*|^2}{2} + \frac{|\tilde{W}_{g22}|^2}{2} + \frac{|W_{g22}^*|^2}{2} + \dots + \\
& + \frac{|\tilde{W}_{g_{n_c2}}|^2}{2} + \frac{|W_{g_{n_c2}}^*|^2}{2} + \dots + \frac{|\tilde{W}_{g1n}|^2}{2} + \frac{|W_{g1n}^*|^2}{2} + \frac{|\tilde{W}_{g2n}|^2}{2} + \frac{|W_{g2n}^*|^2}{2} + \dots + \frac{|\tilde{W}_{g_{n_cn}}|^2}{2} + \frac{|W_{g_{n_cn}}^*|^2}{2} + \\
& + \tilde{x}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u})
\end{aligned} \tag{3.128}$$

We put as:

$$\begin{aligned}
q = & \frac{|\tilde{W}_{f11}|^2}{2} + \frac{|W_{f11}^*|^2}{2} + \frac{|\tilde{W}_{f21}|^2}{2} + \frac{|W_{f21}^*|^2}{2} + \dots + \frac{|\tilde{W}_{f_{n_c1}}|^2}{2} + \frac{|W_{f_{n_c1}}^*|^2}{2} + \frac{|\tilde{W}_{f12}|^2}{2} + \frac{|W_{f12}^*|^2}{2} + \frac{|\tilde{W}_{f22}|^2}{2} + \frac{|W_{f22}^*|^2}{2} + \dots + \\
& + \frac{|\tilde{W}_{f_{n_c2}}|^2}{2} + \frac{|W_{f_{n_c2}}^*|^2}{2} + \dots + \frac{|\tilde{W}_{f1n}|^2}{2} + \frac{|W_{f1n}^*|^2}{2} + \frac{|\tilde{W}_{f2n}|^2}{2} + \frac{|W_{f2n}^*|^2}{2} + \dots + \frac{|\tilde{W}_{f_{n_cn}}|^2}{2} + \frac{|W_{f_{n_cn}}^*|^2}{2} + \\
& + \frac{|\tilde{W}_{g11}|^2}{2} + \frac{|W_{g11}^*|^2}{2} + \frac{|\tilde{W}_{g21}|^2}{2} + \frac{|W_{g21}^*|^2}{2} + \dots + \frac{|\tilde{W}_{g_{n_c1}}|^2}{2} + \frac{|W_{g_{n_c1}}^*|^2}{2} + \frac{|\tilde{W}_{g12}|^2}{2} + \frac{|W_{g12}^*|^2}{2} + \frac{|\tilde{W}_{g22}|^2}{2} + \frac{|W_{g22}^*|^2}{2} + \dots + \\
& + \frac{|\tilde{W}_{g_{n_c2}}|^2}{2} + \frac{|W_{g_{n_c2}}^*|^2}{2} + \dots + \frac{|\tilde{W}_{g1n}|^2}{2} + \frac{|W_{g1n}^*|^2}{2} + \frac{|\tilde{W}_{g2n}|^2}{2} + \frac{|W_{g2n}^*|^2}{2} + \dots + \frac{|\tilde{W}_{g_{n_cn}}|^2}{2} + \frac{|W_{g_{n_cn}}^*|^2}{2} +
\end{aligned}$$

So, the above equation (3.128) becomes:

$$(3.128) \Rightarrow \dot{L} = \tilde{x}^T (A - BK^T) \tilde{x} - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + q + \tilde{x}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u}) \tag{3.129}$$

Where $q \geq 0$ a positive quantity. The quantity $\tilde{x}^T (A - BK^T) \tilde{x}$ can be written in the form:

$$\begin{aligned}
\tilde{x}^T (A - BK^T) \tilde{x} &= \tilde{x}^T \left[\left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) + \left(\frac{A - BK^T - (A - BK^T)^T}{2} \right) \right] \tilde{x} = \\
&= \tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} + \tilde{x}^T \left(\frac{A - BK^T - (A - BK^T)^T}{2} \right) \tilde{x} \tag{3.130}
\end{aligned}$$

If a matrix Q is skew symmetric then it holds that: $x^T Q x = 0$ (3.131). So using property (3.131), equation (3.130) becomes: $\tilde{x}^T (A - BK^T) \tilde{x} = \tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x}$ (3.132). Substituting (3.132) into (3.129) we obtain:

$$\dot{L} = \tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + q + \tilde{x}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u}) \quad (3.133)$$

As the matrix K is chosen as $A - BK^T$ to be Hurwitz this means that it has negative eigenvalues. Also it holds by Lyapunov theory that if a matrix W is Hurwitz then the system represented by the equation $\dot{x} = Wx$ is asymptotically stable and the first derivative of Lyapunov function of this is negative. So, as the matrix is $A - BK^T$ Hurwitz, then the system $\dot{x} = (A - BK^T)x$ (3.134) is asymptotically stable and its first derivative of Lyapunov function is negative.

Consider the Lyapunov function for this system: $L = \frac{1}{2} \tilde{x}^T \tilde{x}$ (3.135). As the system (3.134) is asymptotically stable it holds that: $\dot{L} \leq 0 \Rightarrow \tilde{x}^T \dot{\tilde{x}} \leq 0$. Using equation (3.134), the previous equation becomes: $\tilde{x}^T \dot{\tilde{x}} \leq 0 \Rightarrow \tilde{x}^T (A - BK^T) \tilde{x} \leq 0$. Also using equation (3.132), the previous equation becomes:

$$\tilde{x}^T (A - BK^T) \tilde{x} \leq 0 \Rightarrow \tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} \leq 0 \quad (3.136)$$

A square matrix A is called symmetric if it is equal with its transport and the following property holds:

$$A = A^T \quad (3.137)$$

Also, for any symmetric matrix Q it has been proven and holds that:

$$\lambda_{\min}(Q) x^T x \leq x^T Q x \leq \lambda_{\max}(Q) x^T x, \forall x \in C^m \quad (3.138)$$

So, as the matrix $\frac{A - BK^T + (A - BK^T)^T}{2}$ is symmetric it holds that:

$$\lambda_{\min} \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x}^T \tilde{x} \leq \tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} \leq \lambda_{\max} \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x}^T \tilde{x}$$

So it also holds that:

$$-\lambda_{\min} \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x}^T \tilde{x} \geq -\tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} \quad (3.139)$$

$$\text{As } \tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} \leq 0 \text{ it is:}$$

$$\tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} \leq -\tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x}$$

So, using equation(3.139), the above equation becomes:

$$\tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} \leq -\tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} \leq -\lambda_{\min} \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x}^T \tilde{x} \Rightarrow$$

$$\Rightarrow \tilde{x}^T \left(\frac{A - BK^T + (A - BK^T)^T}{2} \right) \tilde{x} \leq -\frac{\lambda_{\min}}{2} (A - BK^T + (A - BK^T)^T) \|\tilde{x}\|^2 \quad (3.140)$$

So, by using the above equation(3.140), the equation (3.133) becomes:

$$(3.133) \Rightarrow$$

$$\begin{aligned} \Rightarrow \dot{L} \leq & -\frac{\lambda_{\min}}{2} (A - BK^T + (A - BK^T)^T) \|\tilde{x}\|^2 - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + q + \\ & + \tilde{x}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u}) \end{aligned} \quad (3.141)$$

By putting $\sigma_{\min} = -\frac{\lambda_{\min}}{2} (A - BK^T + (A - BK^T)^T)$, the above equation (3.141) becomes:

$$\Rightarrow \dot{L} \leq -\sigma_{\min} \|\tilde{x}\|^2 - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + q + \left[\tilde{x}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u}) \right] \quad (3.142)$$

The above equation can be manipulated as:

$$\Rightarrow \dot{L} \leq -\frac{\sigma_{\min}}{2} \|\tilde{x}\|^2 - \frac{\sigma_{\min}}{2n} \|\tilde{x}\|^2 - \dots - \frac{\sigma_{\min}}{2n} \|\tilde{x}\|^2 - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + q + \left[\tilde{x}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u}) \right] \quad (3.143)$$

Quantity n expresses the degrees of freedom of the Lagrangian system as defined in the problem formulation. So there are n quantities $\left(-\frac{\sigma_{\min}}{2n}\|\tilde{x}\|^2\right)$ in the above equation (3.143).

It holds for the **error bounds** of vectors $\underline{\mathcal{E}}_{f(x)}, \underline{u}$ and matrix $\underline{\mathcal{E}}_{g(x)}$ that:

$$\begin{bmatrix} \mathcal{E}_{1f(x)} \\ \mathcal{E}_{2f(x)} \\ \vdots \\ \mathcal{E}_{nf(x)} \end{bmatrix} \leq \begin{bmatrix} \bar{\mathcal{E}}_{1f(x)} \\ \bar{\mathcal{E}}_{2f(x)} \\ \vdots \\ \bar{\mathcal{E}}_{nf(x)} \end{bmatrix} \Rightarrow \begin{bmatrix} \mathcal{E}_{1f(x)} \leq |\mathcal{E}_{1f(x)}| \leq \bar{\mathcal{E}}_{1f(x)} \\ \mathcal{E}_{2f(x)} \leq |\mathcal{E}_{2f(x)}| \leq \bar{\mathcal{E}}_{2f(x)} \\ \vdots \\ \mathcal{E}_{nf(x)} \leq |\mathcal{E}_{nf(x)}| \leq \bar{\mathcal{E}}_{nf(x)} \end{bmatrix} \quad (3.144)$$

Where $\bar{\mathcal{E}}_{1f(x)}, \bar{\mathcal{E}}_{2f(x)}, \dots, \bar{\mathcal{E}}_{nf(x)}$ the error bounds for each element of the vector representing each degree of freedom of the system.

$$\begin{bmatrix} \mathcal{E}_{11g(x)} & \mathcal{E}_{12g(x)} & \cdot & \cdot & \cdot & \mathcal{E}_{1ng(x)} \\ \mathcal{E}_{21g(x)} & \mathcal{E}_{22g(x)} & \cdot & \cdot & \cdot & \mathcal{E}_{2ng(x)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathcal{E}_{n1g(x)} & \mathcal{E}_{n2g(x)} & \cdot & \cdot & \cdot & \mathcal{E}_{nng(x)} \end{bmatrix} \leq \begin{bmatrix} \bar{\mathcal{E}}_{11g(x)} & \bar{\mathcal{E}}_{12g(x)} & \cdot & \cdot & \cdot & \bar{\mathcal{E}}_{1ng(x)} \\ \bar{\mathcal{E}}_{21g(x)} & \bar{\mathcal{E}}_{22g(x)} & \cdot & \cdot & \cdot & \bar{\mathcal{E}}_{2ng(x)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \bar{\mathcal{E}}_{n1g(x)} & \bar{\mathcal{E}}_{n2g(x)} & \cdot & \cdot & \cdot & \bar{\mathcal{E}}_{nng(x)} \end{bmatrix} \Rightarrow$$

$$\Rightarrow \begin{bmatrix} \mathcal{E}_{11g(x)} \leq |\mathcal{E}_{11g(x)}| \leq \bar{\mathcal{E}}_{11g(x)} & \mathcal{E}_{12g(x)} \leq |\mathcal{E}_{12g(x)}| \leq \bar{\mathcal{E}}_{12g(x)} & \cdot & \cdot & \cdot & \mathcal{E}_{1ng(x)} \leq |\mathcal{E}_{1ng(x)}| \leq \bar{\mathcal{E}}_{1ng(x)} \\ \mathcal{E}_{21g(x)} \leq |\mathcal{E}_{21g(x)}| \leq \bar{\mathcal{E}}_{21g(x)} & \mathcal{E}_{22g(x)} \leq |\mathcal{E}_{22g(x)}| \leq \bar{\mathcal{E}}_{22g(x)} & \cdot & \cdot & \cdot & \mathcal{E}_{2ng(x)} \leq |\mathcal{E}_{2ng(x)}| \leq \bar{\mathcal{E}}_{2ng(x)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathcal{E}_{n1g(x)} \leq |\mathcal{E}_{n1g(x)}| \leq \bar{\mathcal{E}}_{n1g(x)} & \mathcal{E}_{n2g(x)} \leq |\mathcal{E}_{n2g(x)}| \leq \bar{\mathcal{E}}_{n2g(x)} & \cdot & \cdot & \cdot & \mathcal{E}_{nng(x)} \leq |\mathcal{E}_{nng(x)}| \leq \bar{\mathcal{E}}_{nng(x)} \end{bmatrix} \quad (3.145)$$

Where $\bar{\mathcal{E}}_{11g(x)}, \dots, \bar{\mathcal{E}}_{nng(x)}$ the error bounds for each element of the matrix each degree of freedom of the system. For the control signals vector as defined in the equation (3.94):

$$|\underline{u}| \leq \underline{\bar{u}} \Rightarrow \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \leq \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \vdots \\ \bar{u}_n \end{bmatrix} \Rightarrow \begin{bmatrix} u_1 \leq |u_1| \leq \bar{u}_1 \\ u_2 \leq |u_2| \leq \bar{u}_2 \\ \vdots \\ u_n \leq |u_n| \leq \bar{u}_n \end{bmatrix} \quad (3.146)$$

Also as the vector $\tilde{x}^T$ is of dimensions $1 \times n$ as defined in problem formulation and also matrix B is of dimensions $2n \times n$ as defined in the beginning of subsection 3-C. The quantity $\tilde{x}^T B$ can be written as:

$$\tilde{x}^T B = \begin{bmatrix} \tilde{x}_1 & \tilde{x}_2 & \dots & \tilde{x}_n & \dot{\tilde{x}}_1 & \dot{\tilde{x}}_2 & \dots & \dot{\tilde{x}}_n \end{bmatrix} \begin{bmatrix} 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \vdots \\ 1 & 0 & \dots & \vdots \\ 0 & 1 & 0 & \dots \\ \vdots & 0 & 1 & 0 \\ \vdots & \vdots & 0 & 1 \\ 0 & \vdots & \vdots & 0 & 1 \end{bmatrix} = \begin{bmatrix} \dot{\tilde{x}}_1 & \dot{\tilde{x}}_2 & \dots & \dot{\tilde{x}}_n \end{bmatrix} \quad (3.147)$$

So, using (3.147) the quantity $\left[\tilde{x}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u}) \right]$ can be written in the form:

$$\left[\tilde{x}^T B (\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u}) \right] = \begin{bmatrix} \dot{\tilde{x}}_1 & \dot{\tilde{x}}_2 & \dots & \dot{\tilde{x}}_n \end{bmatrix} \left[\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u} \right] \quad (3.148)$$

As it holds that $\underline{\varepsilon}_{f(x)} = \begin{bmatrix} \varepsilon_{1f(x)} \\ \varepsilon_{2f(x)} \\ \vdots \\ \varepsilon_{nf(x)} \end{bmatrix}$, $\underline{\varepsilon}_{g(x)} = \begin{bmatrix} \varepsilon_{11g(x)} & \varepsilon_{12g(x)} & \dots & \varepsilon_{1ng(x)} \\ \varepsilon_{21g(x)} & \varepsilon_{22g(x)} & \dots & \varepsilon_{2ng(x)} \\ \vdots & \vdots & \vdots & \vdots \\ \varepsilon_{n1g(x)} & \varepsilon_{n2g(x)} & \dots & \varepsilon_{nng(x)} \end{bmatrix}$, $\underline{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$ the above equation

(3.148) becomes:

$$\begin{aligned}
& \left[\tilde{\underline{x}}^T B \left(\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u} \right) \right] = \begin{bmatrix} \dot{\tilde{x}}_1 & \dot{\tilde{x}}_2 & \dots & \dots & \dot{\tilde{x}}_n \end{bmatrix} \left(\begin{bmatrix} \varepsilon_{1f(x)} \\ \varepsilon_{2f(x)} \\ \vdots \\ \vdots \\ \varepsilon_{nf(x)} \end{bmatrix} + \begin{bmatrix} \varepsilon_{11g(x)} & \varepsilon_{12g(x)} & \dots & \dots & \varepsilon_{1ng(x)} \\ \varepsilon_{21g(x)} & \varepsilon_{22g(x)} & \dots & \dots & \varepsilon_{2ng(x)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{n1g(x)} & \varepsilon_{n2g(x)} & \dots & \dots & \varepsilon_{nng(x)} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_n \end{bmatrix} \right) \Rightarrow \\
& \Rightarrow \left[\tilde{\underline{x}}^T B \left(\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u} \right) \right] = \begin{bmatrix} \dot{\tilde{x}}_1 & \dot{\tilde{x}}_2 & \dots & \dots & \dot{\tilde{x}}_n \end{bmatrix} \begin{bmatrix} \varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \\ \varepsilon_{2f(x)} + \varepsilon_{21g(x)}u_1 + \varepsilon_{22g(x)}u_2 + \dots + \varepsilon_{2ng(x)}u_n \\ \vdots \\ \vdots \\ \varepsilon_{nf(x)} + \varepsilon_{n1g(x)}u_1 + \varepsilon_{n2g(x)}u_2 + \dots + \varepsilon_{nng(x)}u_n \end{bmatrix} \Rightarrow \\
& \Rightarrow \left[\tilde{\underline{x}}^T B \left(\underline{\varepsilon}_{f(x)} + \underline{\varepsilon}_{g(x)} \underline{u} \right) \right] = \dot{\tilde{x}}_1 \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right) + \\
& + \dot{\tilde{x}}_2 \left(\varepsilon_{2f(x)} + \varepsilon_{21g(x)}u_1 + \varepsilon_{22g(x)}u_2 + \dots + \varepsilon_{2ng(x)}u_n \right) + \dots + \dot{\tilde{x}}_n \left(\varepsilon_{nf(x)} + \varepsilon_{n1g(x)}u_1 + \varepsilon_{n2g(x)}u_2 + \dots + \varepsilon_{nng(x)}u_n \right) \quad (3.149)
\end{aligned}$$

Using equation (3.149) then (3.143) becomes:

$$\begin{aligned}
\dot{L} \leq & -\frac{\sigma_{\min}}{2} \|\tilde{\underline{x}}\|^2 - \frac{\sigma_{\min}}{2n} \|\tilde{\underline{x}}\|^2 - \dots - \frac{\sigma_{\min}}{2n} \|\tilde{\underline{x}}\|^2 - \sigma_f \text{trace}(\tilde{W}_f^T \tilde{W}_f) - \sigma_g \text{trace}(\tilde{W}_g^T \tilde{W}_g) + q + \\
& + \dot{\tilde{x}}_1 \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right) + \\
& + \dot{\tilde{x}}_2 \left(\varepsilon_{2f(x)} + \varepsilon_{21g(x)}u_1 + \varepsilon_{22g(x)}u_2 + \dots + \varepsilon_{2ng(x)}u_n \right) + \dots + \dot{\tilde{x}}_n \left(\varepsilon_{nf(x)} + \varepsilon_{n1g(x)}u_1 + \varepsilon_{n2g(x)}u_2 + \dots + \varepsilon_{nng(x)}u_n \right) \quad (3.150)
\end{aligned}$$

The quantity $-\frac{\sigma_{\min}}{2n} \|\tilde{\underline{x}}\|^2 + \dot{\tilde{x}}_1 \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right)$ can be written in the form:

$$\begin{aligned}
& -\frac{\sigma_{\min}}{2n} \|\tilde{\underline{x}}\|^2 + \dot{\tilde{x}}_1 \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right) = \\
& -\frac{1}{2} \left(\sqrt{\frac{\sigma_{\min}}{n}} \|\tilde{\underline{x}}\| \right)^2 + \sqrt{\frac{\sigma_{\min}}{n}} \frac{\dot{\tilde{x}}_1 \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right)}{\sqrt{\frac{\sigma_{\min}}{n}}} \quad (3.151)
\end{aligned}$$

It holds that: $\dot{\tilde{x}}_1 \leq |\dot{\tilde{x}}_1| = \sqrt{\dot{\tilde{x}}_1^2} \leq \sqrt{\tilde{x}_1^2 + \tilde{x}_2^2 + \dots + \tilde{x}_n^2 + \dot{\tilde{x}}_1^2 + \dot{\tilde{x}}_2^2 + \dots + \dot{\tilde{x}}_n^2} = \|\tilde{\underline{x}}\|$. So using this inequality, the above inequality (3.151) becomes:

$$\begin{aligned}
& -\frac{\sigma_{\min}}{2n} \|\tilde{\underline{x}}\|^2 + \dot{\tilde{\chi}}_1 \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right) = \\
& -\frac{1}{2} \left(\sqrt{\frac{\sigma_{\min}}{n}} \|\tilde{\underline{x}}\| \right)^2 + \sqrt{\frac{\sigma_{\min}}{n}} \frac{\dot{\tilde{\chi}}_1 \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right)}{\sqrt{\frac{\sigma_{\min}}{n}}} \leq \quad (3.152) \\
& \leq -\frac{1}{2} \left(\sqrt{\frac{\sigma_{\min}}{n}} \|\tilde{\underline{x}}\| \right)^2 + \sqrt{\frac{\sigma_{\min}}{n}} \frac{\|\tilde{\underline{x}}\| \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right)}{\sqrt{\frac{\sigma_{\min}}{n}}}
\end{aligned}$$

Using the Young's inequality(3.125), the above inequality becomes:

(3.152) $\Rightarrow$

$$\begin{aligned}
& \Rightarrow -\frac{1}{2} \left(\sqrt{\frac{\sigma_{\min}}{n}} \|\tilde{\underline{x}}\| \right)^2 + \sqrt{\frac{\sigma_{\min}}{n}} \frac{\|\tilde{\underline{x}}\| \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right)}{\sqrt{\frac{\sigma_{\min}}{n}}} \leq \\
& \leq \frac{1}{2} \left(\frac{\left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right)^2}{\sqrt{\frac{\sigma_{\min}}{n}}} \right)^2 = \frac{n \left(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n \right)^2}{2\sigma_{\min}} \quad (3.153)
\end{aligned}$$

Doing the same steps for the quantity $-\frac{\sigma_{\min}}{2n} \|\tilde{\underline{x}}\|^2 + \dot{\tilde{\chi}}_2 \left(\varepsilon_{2f(x)} + \varepsilon_{21g(x)}u_1 + \varepsilon_{22g(x)}u_2 + \dots + \varepsilon_{2ng(x)}u_n \right)$ it holds that:

$$\begin{aligned}
& -\frac{\sigma_{\min}}{2n} \|\tilde{\underline{x}}\|^2 + \dot{\tilde{\chi}}_2 \left(\varepsilon_{2f(x)} + \varepsilon_{21g(x)}u_1 + \varepsilon_{22g(x)}u_2 + \dots + \varepsilon_{2ng(x)}u_n \right) \leq \frac{n \left(\varepsilon_{2f(x)} + \varepsilon_{21g(x)}u_1 + \varepsilon_{22g(x)}u_2 + \dots + \varepsilon_{2ng(x)}u_n \right)^2}{2\sigma_{\min}} \\
& (3.154)
\end{aligned}$$

Also for the quantity $-\frac{\sigma_{\min}}{2n} \|\tilde{\underline{x}}\|^2 + \dot{\tilde{\chi}}_n \left(\varepsilon_{nf(x)} + \varepsilon_{n1g(x)}u_1 + \varepsilon_{n2g(x)}u_2 + \dots + \varepsilon_{nng(x)}u_n \right)$ it holds that:

$$\begin{aligned}
& -\frac{\sigma_{\min}}{2n} \|\tilde{\underline{x}}\|^2 + \dot{\tilde{\chi}}_n \left(\varepsilon_{nf(x)} + \varepsilon_{n1g(x)}u_1 + \varepsilon_{n2g(x)}u_2 + \dots + \varepsilon_{nng(x)}u_n \right) \leq \frac{n \left(\varepsilon_{nf(x)} + \varepsilon_{n1g(x)}u_1 + \varepsilon_{n2g(x)}u_2 + \dots + \varepsilon_{nng(x)}u_n \right)^2}{2\sigma_{\min}} \\
& (3.155)
\end{aligned}$$

Using the inequalities proven above(3.153),(3.154),(3.155) then the equation (3.150) becomes:

$$\begin{aligned}
\dot{L} \leq & -\frac{\sigma_{\min}}{2} \|\tilde{\underline{x}}\|^2 - \sigma_f \text{trace}(\tilde{\underline{W}}_f^T \tilde{\underline{W}}_f) - \sigma_g \text{trace}(\tilde{\underline{W}}_g^T \tilde{\underline{W}}_g) + q + \\
& + \frac{n(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n)^2}{2\sigma_{\min}} + \frac{n(\varepsilon_{2f(x)} + \varepsilon_{21g(x)}u_1 + \varepsilon_{22g(x)}u_2 + \dots + \varepsilon_{2ng(x)}u_n)^2}{2\sigma_{\min}} + \\
& + \frac{n(\varepsilon_{nf(x)} + \varepsilon_{n1g(x)}u_1 + \varepsilon_{n2g(x)}u_2 + \dots + \varepsilon_{nng(x)}u_n)^2}{2\sigma_{\min}}
\end{aligned} \quad (3.156)$$

It can be written that:

$$\begin{aligned}
d = q + & \frac{n(\varepsilon_{1f(x)} + \varepsilon_{11g(x)}u_1 + \varepsilon_{12g(x)}u_2 + \dots + \varepsilon_{1ng(x)}u_n)^2}{2\sigma_{\min}} + \\
& + \frac{n(\varepsilon_{2f(x)} + \varepsilon_{21g(x)}u_1 + \varepsilon_{22g(x)}u_2 + \dots + \varepsilon_{2ng(x)}u_n)^2}{2\sigma_{\min}} + \frac{n(\varepsilon_{nf(x)} + \varepsilon_{n1g(x)}u_1 + \varepsilon_{n2g(x)}u_2 + \dots + \varepsilon_{nng(x)}u_n)^2}{2\sigma_{\min}}
\end{aligned} \quad (3.157)$$

Where $d \geq 0$ a positive quantity. Then, (3.156) becomes:

$$\dot{L} \leq -\frac{\sigma_{\min}}{2} \|\tilde{\underline{x}}\|^2 - \sigma_f \text{trace}(\tilde{\underline{W}}_f^T \tilde{\underline{W}}_f) - \sigma_g \text{trace}(\tilde{\underline{W}}_g^T \tilde{\underline{W}}_g) + d \quad (3.158)$$

So, $\dot{L} < 0$ when:

$$\frac{\sigma_{\min}}{2} \|\tilde{\underline{x}}\|^2 > d \Rightarrow \|\tilde{\underline{x}}\| > \sqrt{\frac{2d}{\sigma_{\min}}} \quad (3.159)$$

$$\sigma_f \text{trace}(\tilde{\underline{W}}_f^T \tilde{\underline{W}}_f) > d \Rightarrow \text{trace}(\tilde{\underline{W}}_f^T \tilde{\underline{W}}_f) > \frac{d}{\sigma_f} \quad (3.160)$$

$$\sigma_g \text{trace}(\tilde{\underline{W}}_g^T \tilde{\underline{W}}_g) > d \Rightarrow \text{trace}(\tilde{\underline{W}}_g^T \tilde{\underline{W}}_g) > \frac{d}{\sigma_g} \quad (3.161)$$

This leads to the uniform ultimate boundness (UUB) of $\tilde{\underline{x}}, \tilde{\underline{W}}_f^T, \tilde{\underline{W}}_g^T$ with respect to the sets:

$$X = \left\{ \tilde{\underline{x}} \in \mathfrak{R}^n : \|\tilde{\underline{x}}\| \leq \sqrt{\frac{2d}{\sigma_{\min}}} \right\}, W_f = \left\{ \tilde{\underline{W}}_f \in \mathfrak{R}^{pf} : \text{trace}(\tilde{\underline{W}}_f^T \tilde{\underline{W}}_f) > \frac{d}{\sigma_f} \right\}, W_g = \left\{ \tilde{\underline{W}}_g \in \mathfrak{R}^{pf} : \text{trace}(\tilde{\underline{W}}_g^T \tilde{\underline{W}}_g) > \frac{d}{\sigma_g} \right\}$$

Thus, $\tilde{\underline{x}}, \tilde{\underline{W}}_f, \tilde{\underline{W}}_g \in L_\infty$. Additionally it has been proven in the previous subsection that under the excitation of (3.76) the system (2.62) generates uniformly bounded state trajectories. Hence, $\hat{\underline{x}} = \underline{x} - \tilde{\underline{x}} \in L_\infty$. Therefore all signals of the identification scheme remain bounded. So it can be verified that $\|\tilde{\underline{x}}(t)\|$ converges to a neighborhood of zero whose size is regulated by permitting

$\sigma_{\min}$ to admit sufficiently large values by choosing appropriately the K matrix and putting small

values for σ_f, σ_g . Also, according to the Proposition 2 (made above), the regressor vectors $Z_f(x), Z_g(x)$ are persistently exciting. So, using the convergence analysis derived from the equations (3.113), (3.114), the parametric errors $\tilde{W}_f, \tilde{W}_g$ converge to neighborhoods of zero, whose size is determined by the value of d which is closely influenced by the values of parameters $\sigma_{\min}, \sigma_f, \sigma_g$ (to regulate the size of $\|\tilde{x}(t)\|$) for a given RBF network structure which corresponds to specific modeling error bounds $\bar{\varepsilon}_f, \bar{\varepsilon}_g$.

CHAPTER 4 – SIMULATION RESULTS

For the verification of the RBF identification scheme proposed above, different simulation examples are presented and the identification process applied is assessed by the graphs produced. As this work is based in the identification of Lagrange systems, firstly we consider a system written in Lagrange dynamics equation. The technique proposed in this work in chapter three is based in systems in canonical form, so the Lagrange system is then transformed to a system in canonical form. Three different examples are simulated in this chapter. Firstly we consider a Lagrange system of one degree of freedom. The nonlinear parts of this Lagrange system are chosen in a way as the corresponding canonical form has in a first case a nonlinear $f(x)$ function and $g(x)$ a constant and known function, and in the second case both $f(x), g(x)$ are unknown nonlinear functions to be identified by the proposed technique of this work. These two examples are analyzed and shown in the following part of this section. Also, both these simulations are accompanied with graphs showing the quality of the learning result. (A third example is also appeared in which, a second order Lagrange system of two degrees of freedom is proposed whose nonlinear parts are identified using the identification technique of this work).

Example 1 – Second order Lagrange system of one degree of freedom with constant $g(x)$

A second order Lagrange system of one degree of freedom is proposed which takes the following form:

$$I\ddot{q} + ml \sin q (\dot{q})^2 = u \quad (4.1)$$

Where I, m, l are system's constant parameters denoted as:

I ...moment of inertia

m ...mass

l ...length

The above Lagrangian equation can be transformed in a system of canonical form. If we put: x_1 as the position and x_2 the velocity of the system, then we have:

$$x_1 = q$$

$$x_2 = \dot{q}$$

So, the above Lagrangian equation can be transformed to:

$$\ddot{q} = -\frac{ml}{I} \sin q (\dot{q})^2 + \frac{1}{I} u \quad (4.2)$$

Using the equations $x_1 = q$ and $x_2 = \dot{q}$ the system in canonical form is:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{ml}{I} \sin x_1 (x_2)^2 + \frac{1}{I} u$$

The identification scheme developed in this worked was based in systems of canonical form with general equation as shown in equation(2.62). In the case of this simulation, the system is second order and there is one degree of freedom so the general form of (2.62) is transformed to:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = f(x) + g(x) \cdot u \quad \text{with } x_1(0) = x_2(0) = 0$$

So, it is:

$$f(x) = -\frac{ml}{I} \sin x_1 (x_2)^2 \quad (4.3)$$

$$g(x) = \frac{1}{I} \quad (4.4)$$

The parameters chosen in this example are:

$$I = 1, \quad m = 0.333, \quad l = 0.65$$

So the equation $f(x)$ which is going to be identified is:

$$f(x_1, x_2) = -0.21645 \sin(x_1) \cdot x_2^2 \quad (4.5)$$

The identification scheme presented in chapter four is used for local identification and in this example the identification is selected to be done in the following area:

$$-1.5 \leq x_1 \leq 1.5$$

$$-1.5 \leq x_2 \leq 1.5$$

According to the identification methodology the nonlinear function $f(x)$ is approximated by the following linear form:

$$f(x) = W_f^{*T} Z_f(x) + \varepsilon_f(x) \quad (4.6)$$

$\varepsilon_f(x)$... is the error term

W_f^{*T} ... is the optimal weight vector

$Z_f(x)$... is the regressor vector

The elements of the regressor vector $Z_f(x) = [zf_1 \quad \dots \quad zf_n]^T$ where $Z_f(x) \in \mathfrak{R}^n$ are calculated from the following equation:

$$zf_i(x) = \exp\left(-\left\|\frac{x - cf_i}{\sigma}\right\|^2\right) \quad (4.7), \quad i = 1, \dots, n$$

Where n is the number of nodes placed in the compact set.

cf_i ... is the vector of the coordinates of each of the selected nodes inside the compact set

Also in the above calculation we have to select the width σ . According to the methodology proposed in chapter two, after the compact set is defined, where the learning is attempted to be made, then we select the number of nodes from which the desired route is passed. This desired route is defined from a set of ellipses while a different ellipse is passed through each node. Each ellipse's parameters are calculated using an optimization algorithm mentioned in chapter two.

After the desired trajectory is defined in the form $\bar{x}_d(t) = [x_d(t), \dot{x}_d(t)]^T$, then the parameters of the adaptive controller should be selected in order to ensure that the system states follow the desired trajectory. We select ϵ such that $\epsilon < \min_{i \neq j} \left\{ \frac{1}{2} \|cf_i - cf_j\| \right\}$. It is $\min_{i \neq j} \left\{ \frac{1}{2} \|cf_i - cf_j\| \right\} = 0.5$. So we

select $\epsilon = 0.5 \min_{i \neq j} \left\{ \frac{1}{2} \|cf_i - cf_j\| \right\} = 0.25$. We also select $\lambda = 1.5, k = 12.5$. In addition, as the quantity

$\lim_{t \rightarrow \infty} \rho_\epsilon(t)$ must satisfy the condition $\lim_{t \rightarrow \infty} \rho_\epsilon(t) < \frac{\epsilon}{2\sqrt{\sum_{i=1}^n \left(\frac{2^{i-1}}{\lambda^{n-i}}\right)^2}}$, it is selected that

$\lim_{t \rightarrow \infty} \rho_\epsilon(t) = 0.5 \frac{\epsilon}{2\sqrt{\sum_{i=1}^n \left(\frac{2^{i-1}}{\lambda^{n-i}}\right)^2}} = 0.0419$. We choose also $\rho_\epsilon(0)$ such that it satisfies the

condition $\rho_\varepsilon(0) > |\varepsilon(0)|$. This can be done by choosing $\rho_\varepsilon(0) = 2 * |\varepsilon(0)| + 0.1 = 2.3757$. By the above selections of controller parameters, it is guaranteed that the vector $Z_f(x(t))$ is persistently exciting and the performance function is:

$$\rho_\varepsilon(t) = (\rho_\varepsilon(0) - \rho_\infty(t))e^{-t} + \rho_\infty(t) = 2.3338e^{-t} + 0.0419 \quad (4.8)$$

Simulation with N=16 nodes

At first step we run a simulation choosing 16 nodes inside the predefined compact set. Figure 7 shows the phase plane trajectory of the system state $x(t)$ along with the desired one $\bar{x}_d(t)$ as well as the centers of the Gaussian basis functions which are the selected nodes inside the compact set.

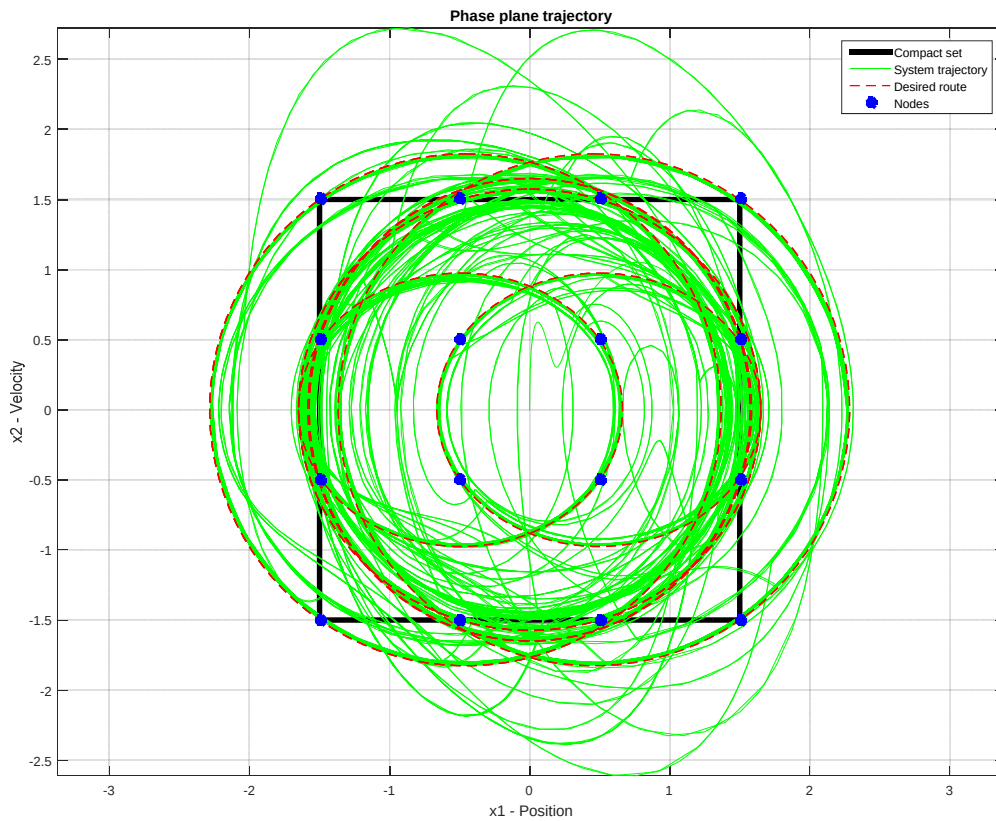


Figure 6. Phase plane trajectory of the system state $x(t)$ with green solid line and the desired one with red dashed line. The centers of the Gaussian basis functions are appeared in blue color.

The desired trajectory of position and velocity is also plotted below as a function of time along with the system states showing that the controller action guarantees that system states follow successfully the desired trajectory.

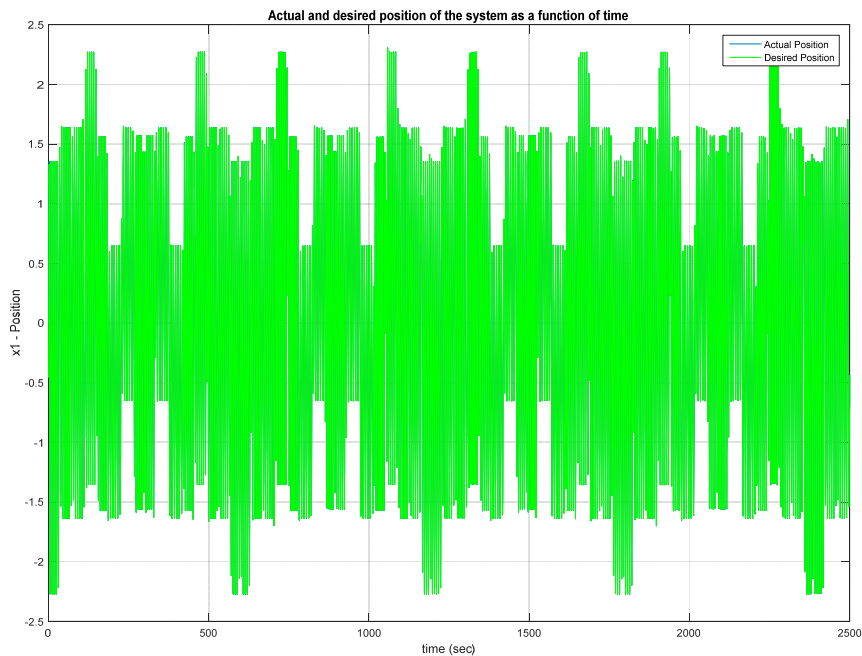


Figure 7. Actual and desired values of x_1 -position of the system as a function of time

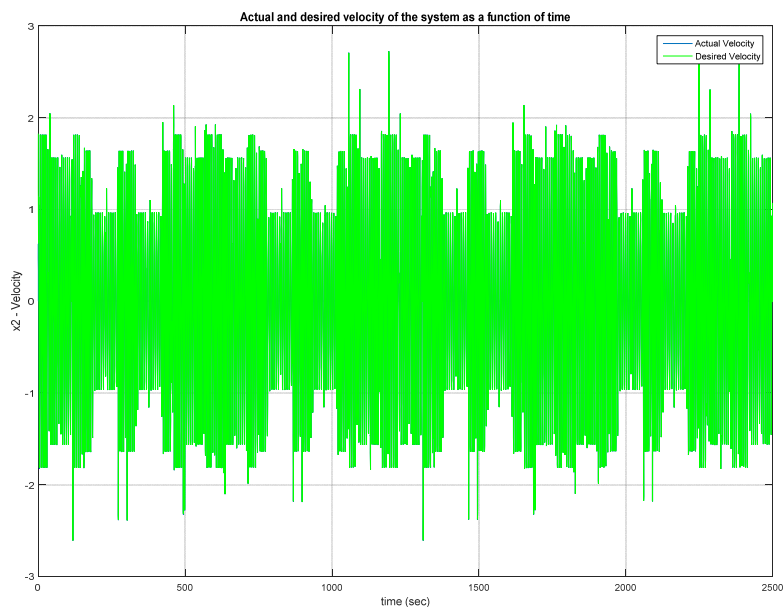


Figure 8. Actual and desired values of x_2 -velocity of the system as a function of time

From the above figures 7 and 8, it is clear that, the controller guarantees that system states $x_1(t), x_2(t)$ follow the desired trajectory with insignificant error. The values of control action are also appeared as a function of time.

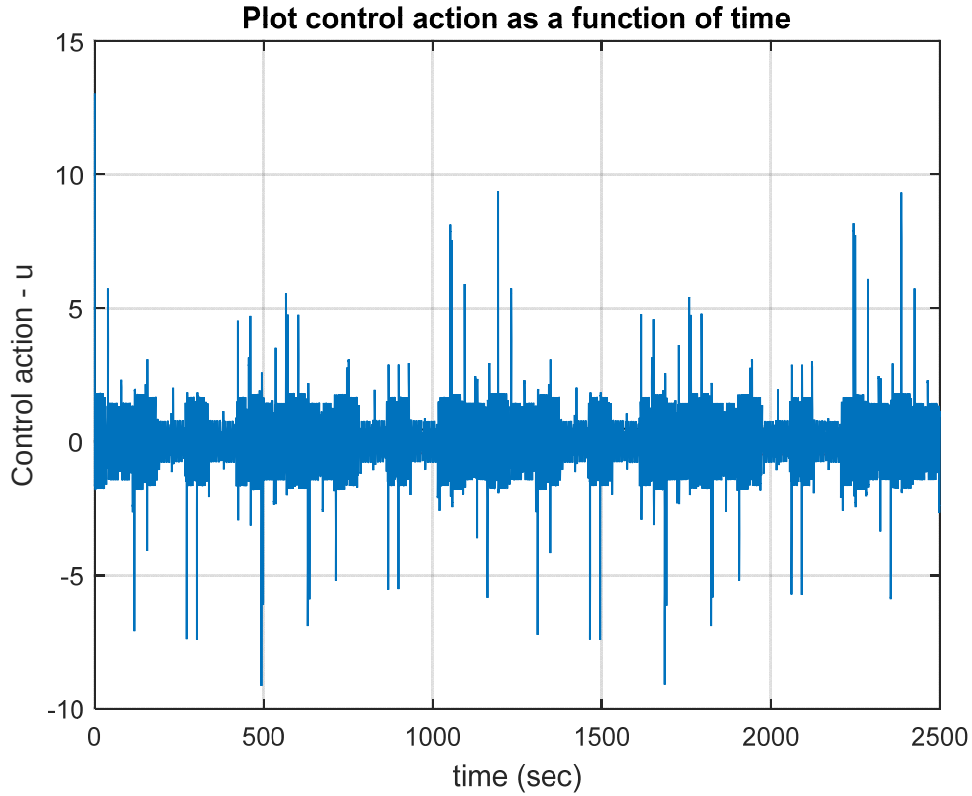


Figure 9. Control action which guarantees that system states follow the desired trajectory

By utilizing the proposed control input, the produced orbit visits periodically an ϵ – neighborhood of the Gaussian centers as it is appeared in the figure 7.

The identification scheme (3.102)-(3.104) was simulated by selecting appropriately the parameters $\gamma_f, \gamma_g, \sigma_f, \sigma_g$ and also the matrix K used in the equation (3.102) in a way that the matrix $A - BK^T$ is Hurwitz. We select $K = [4 \ 4]^T$. For the simulation with 16 nodes we select the parameters $\gamma_f = 2, \sigma_f = 0.0000001$ and also we select the width of RBFs as $\sigma = 0.55$.

In this simulation of 16 nodes, the optimal weight vector is calculated as:

$$W_f^* = [0.4858 \ 0.2335 \ -0.2335 \ -0.4858 \ -0.054 \ -0.0259 \ 0.0259 \ 0.054 \\ 0.054 \ 0.0259 \ -0.0259 \ -0.054 \ -0.4858 \ -0.2335 \ 0.2335 \ 0.4858]^T$$

The convergence of the weight vector parameters W_f estimate after the end of simulation is presented in the following figure 10.

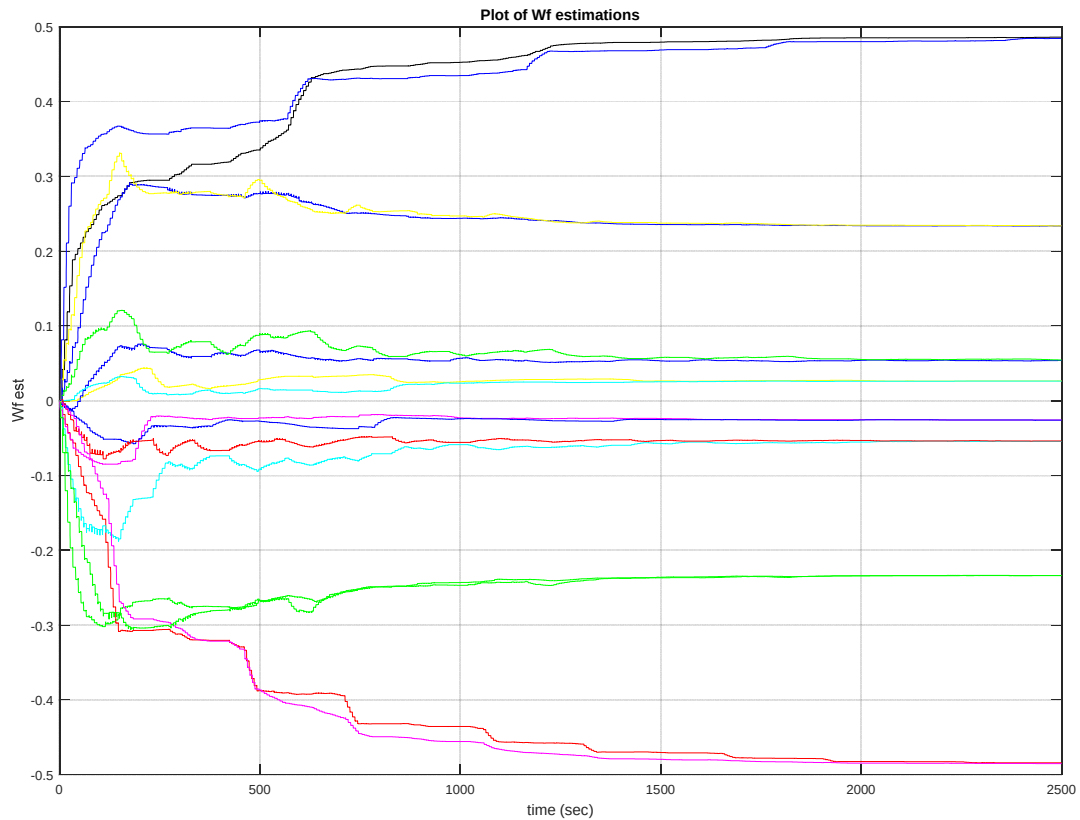


Figure 10. Plot of W_f estimations calculated by the identifier

As it was stated in the theoretical analysis the estimated parameters converge to a small area of their actual (optimal) values achieving the target of learning the unknown nonlinear function $f(x)$. The convergence can be evaluated by calculating the maximum and average estimation error between the W_f estimated values and the actual ones. It is:

Maximum estimation error = $0.0012 = 0.12\%$, Average estimation error = $2.9857 \times 10^{-4} = 0.02957\%$

Also the function produced by the identifier as an estimation of the nonlinear function $f(x)$ is plotted in the figure 11 along with the nonlinear function $f(x)$ with the intention of evaluating the learning quality of the proposed technique.

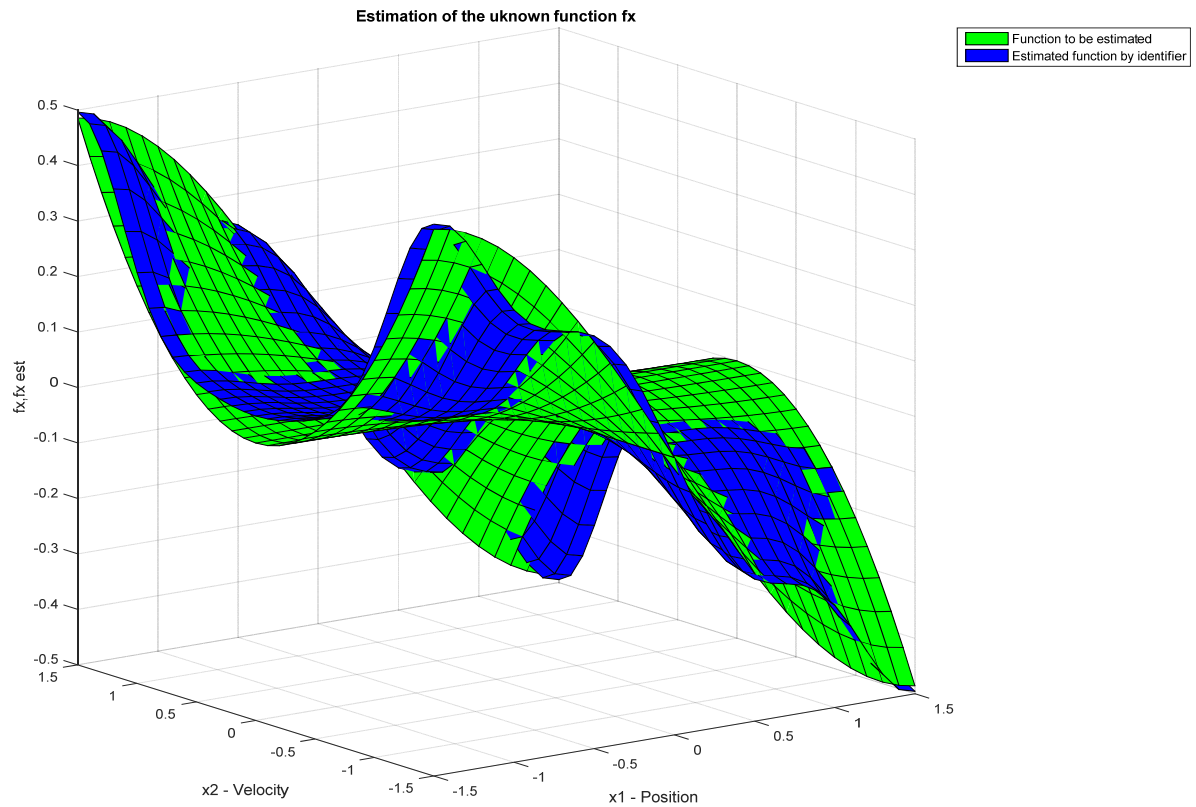


Figure 11. Plot of the function estimated by the identification scheme (selecting 16 nodes) along with the function $f(x)$ intended to be estimated

Another plot is also appeared showing the errors between the real nonlinear function and the estimation of this function.

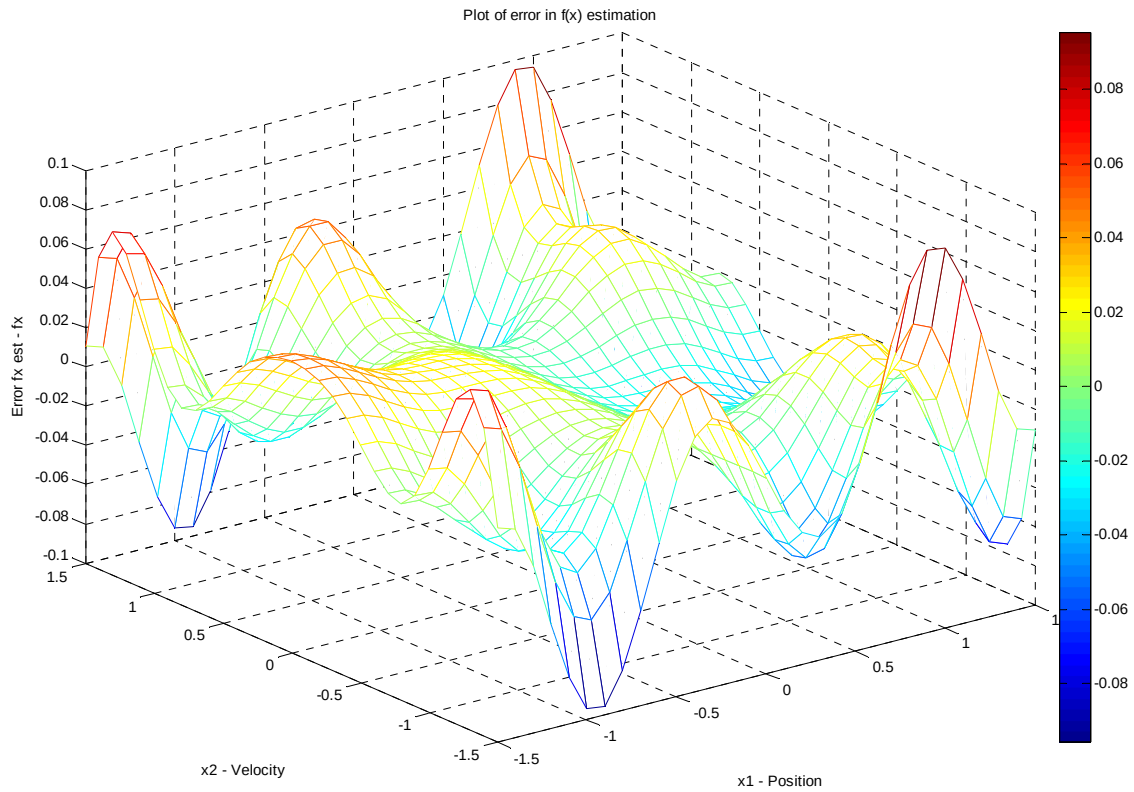


Figure 12. Plot of errors between the function estimated by the identification scheme (selecting 16 nodes) and the function $f(x)$ intended to be estimated

The maximum and average errors calculated are:

Maximum error = $0.0956 = 9.56\%$, Average error = $0.0218 = 2.18\%$

The plot of function $f(x)$ and the estimation of $f(x)$ inside the compact set are also appeared in separate figures below:

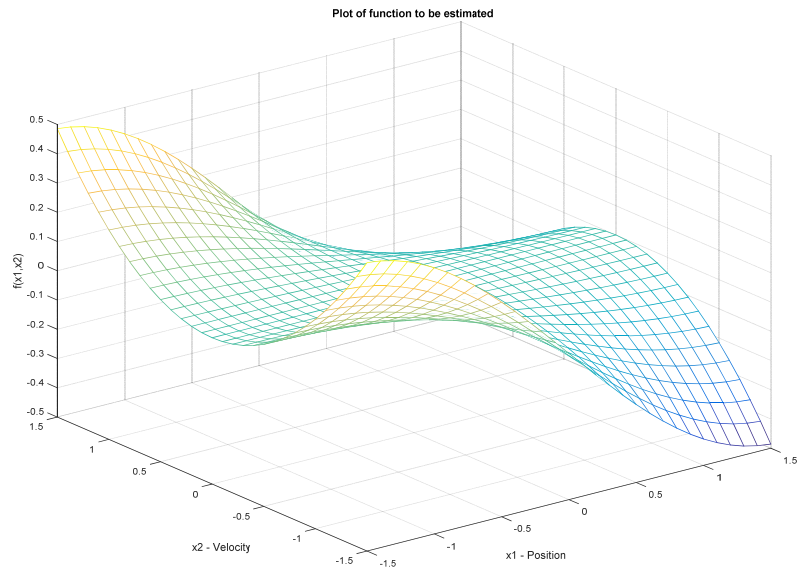


Figure 13. Plot of function $f(x)$ to be estimated by the identification scheme

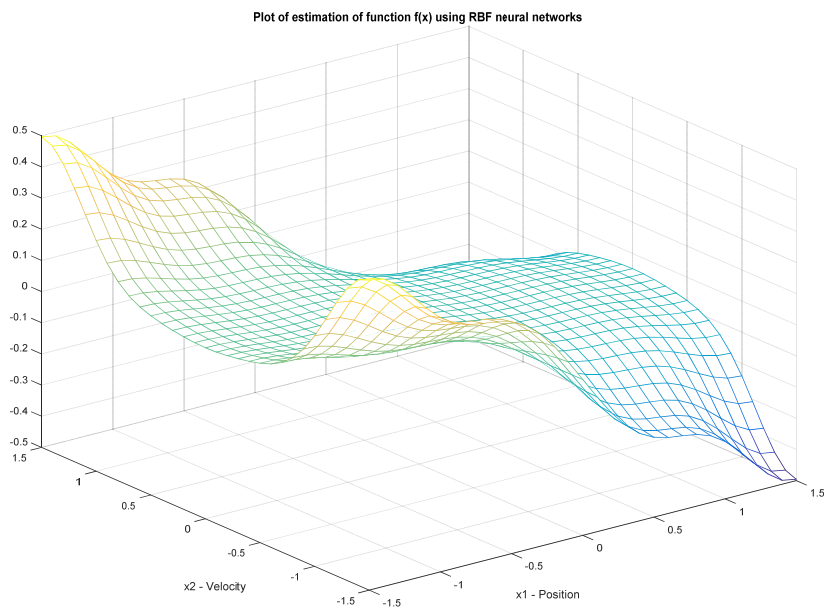


Figure 14. Plot of estimation of function $f(x)$ calculated from the identification scheme selecting 16 nodes inside the compact set

From the above figures 11-14 it is concluded that, the target of learning is achieved producing a local satisfactory approximation of the unknown nonlinear function $f(x)$ in the area of interest where the compact set is defined. The approximation errors shown in the figure 13 can be reduced if the number of the selected nodes is increased. By increasing the number of nodes, more system state values will be stimulated by the learning technique which will be identified by the identifier, having as a drawback the increase of the computation time. In the next simulation we increase the number of selected nodes in order to enhance the results of learning.

Simulation with N=36 nodes

By increasing the number of nodes, more RBF centers are going to be put inside the compact set improving the final result of learning, as more system state values (inside the compact set) are stimulated and are going to be identified by the identifier. The same steps with the previous simulation are followed here. Firstly the phase plane trajectory is appeared in the following figure:

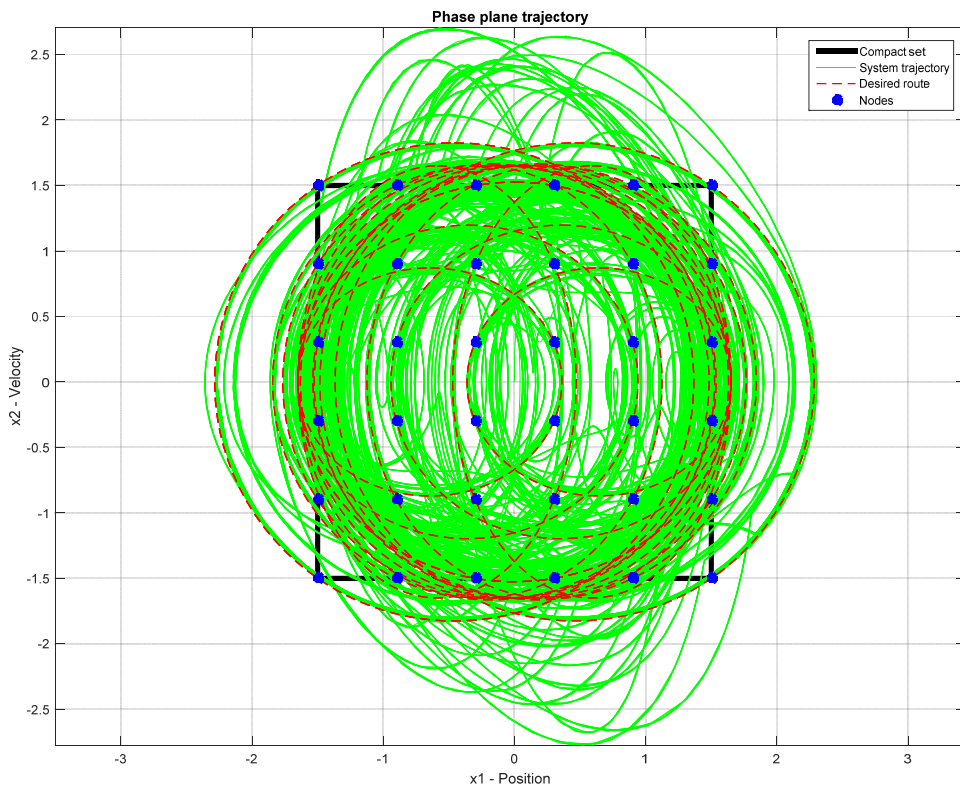


Figure 15. Phase plane trajectory of the system state $x(t)$ with green solid line and the desired one with red dashed line. The centers of the Gaussian basis functions are appeared with blue color

The controller parameters remain the same as in the previous simulation, while the identifier parameters are changed as:

$\gamma_f = 2, \sigma_f = 0.0000001$ and also we select the width of RBFs as $\sigma = 0.33$.

In the case simulated in this section the optimal weight vector is:

$$W_f^* = [0.4858 \quad 0.3815 \quad 0.1439 \quad -0.1439 \quad -0.3815 \quad -0.4858 \quad -0.1749 \quad -0.1373 \\ -0.0518 \quad 0.0518 \quad 0.1373 \quad 0.1749 \quad 0.0194 \quad 0.0153 \quad 0.0058 \quad -0.0058 \\ -0.0153 \quad -0.0194 \quad -0.0194 \quad -0.0153 \quad -0.0058 \quad 0.0058 \quad 0.0153 \quad 0.0194 \\ 0.1749 \quad 0.1373 \quad 0.0518 \quad -0.0518 \quad -0.1373 \quad -0.1749 \quad -0.4858 \quad -0.3815 \\ -0.1439 \quad 0.1439 \quad 0.3815 \quad 0.4858]^T$$

The estimation of the weight vector which is calculated by the identifier is:

$$W_f = [0.4807 \quad 0.3830 \quad 0.1423 \quad -0.1424 \quad -0.3832 \quad -0.4817 \quad -0.1752 \quad -0.1368 \\ -0.0521 \quad 0.0521 \quad 0.1371 \quad 0.1749 \quad 0.0194 \quad 0.0152 \quad 0.0057 \quad -0.0058 \\ -0.0153 \quad -0.0195 \quad -0.0195 \quad -0.0153 \quad -0.0058 \quad 0.0057 \quad 0.0153 \quad 0.0193 \\ 0.1754 \quad 0.1368 \quad 0.0520 \quad -0.0521 \quad -0.1372 \quad -0.1749 \quad -0.4832 \quad -0.3826 \\ -0.1426 \quad 0.1425 \quad 0.3833 \quad 0.4797]^T$$

The approximation achieved by the identifier is quite satisfactory as the estimated values of the weight vector converge to the optimal weight vector values with an insignificant approximation error. These approximation errors are calculated and they are:

Maximum estimation error = $0.0061 = 0.61\%$, Average estimation error = $9.3321 \times 10^{-4} = 0.093321\%$

Also, the convergence of the weight vector parameters W_f estimation is presented in the following figure 16:

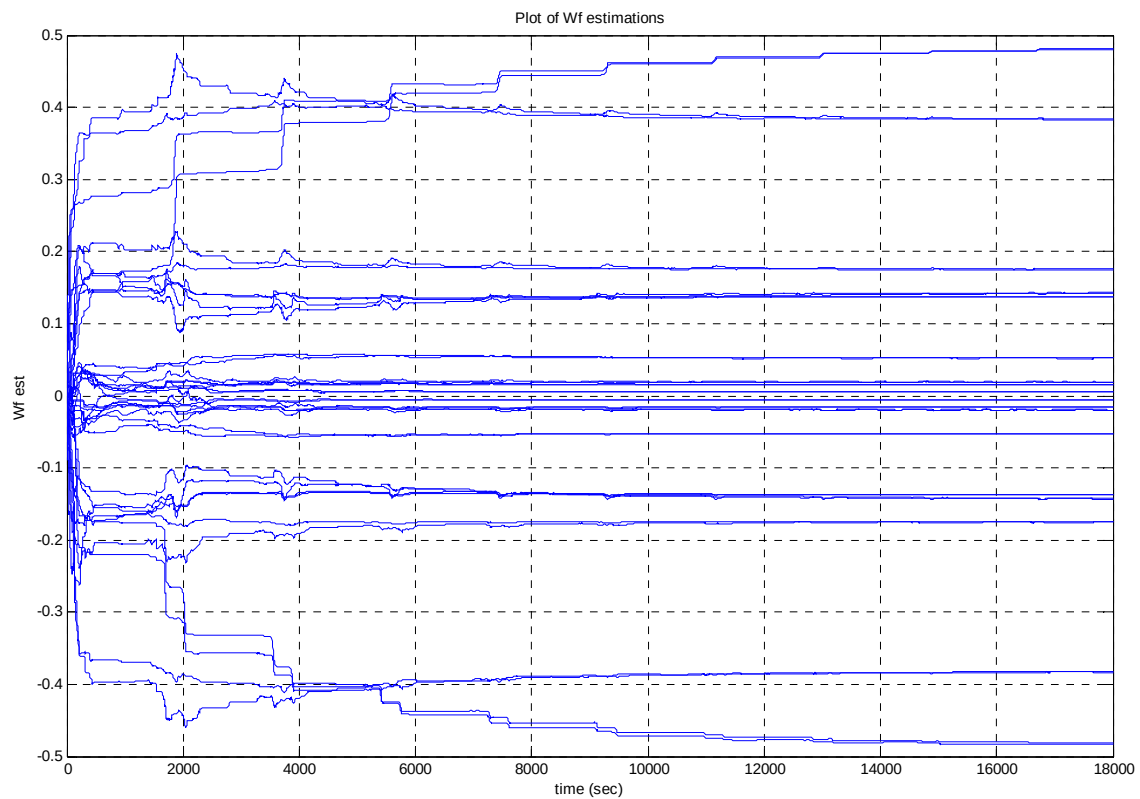


Figure 16. Plot of W_f estimations calculated by the identifier (36 nodes simulation)

The function produced by the identifier as an estimation of the nonlinear function $f(x)$ is plotted in the following figure 16 along with the nonlinear function $f(x)$.

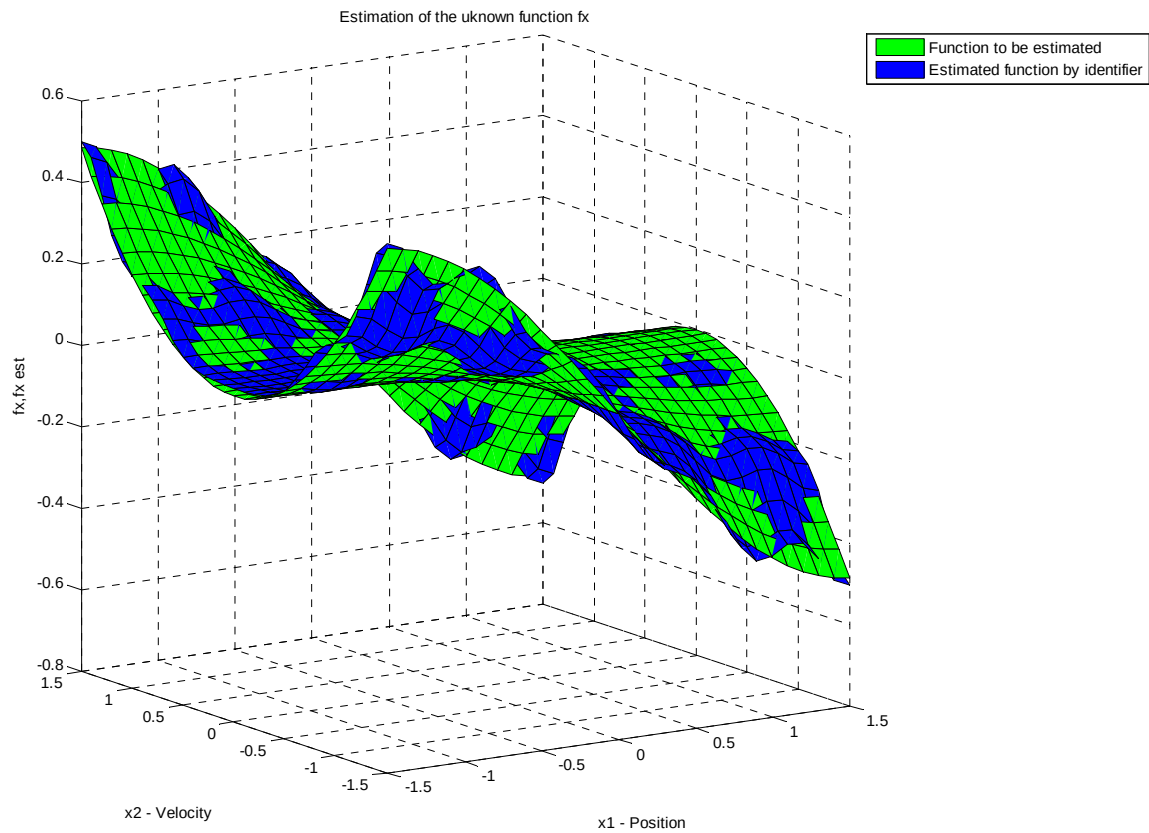


Figure 17. Plot of the function estimated by the identification scheme (selecting 36 nodes) along with the function $f(x)$ intended to be estimated

In the following plot the errors between the estimated non-linear function and the real nonlinear function are appeared giving an evaluation of the learning efficiency.

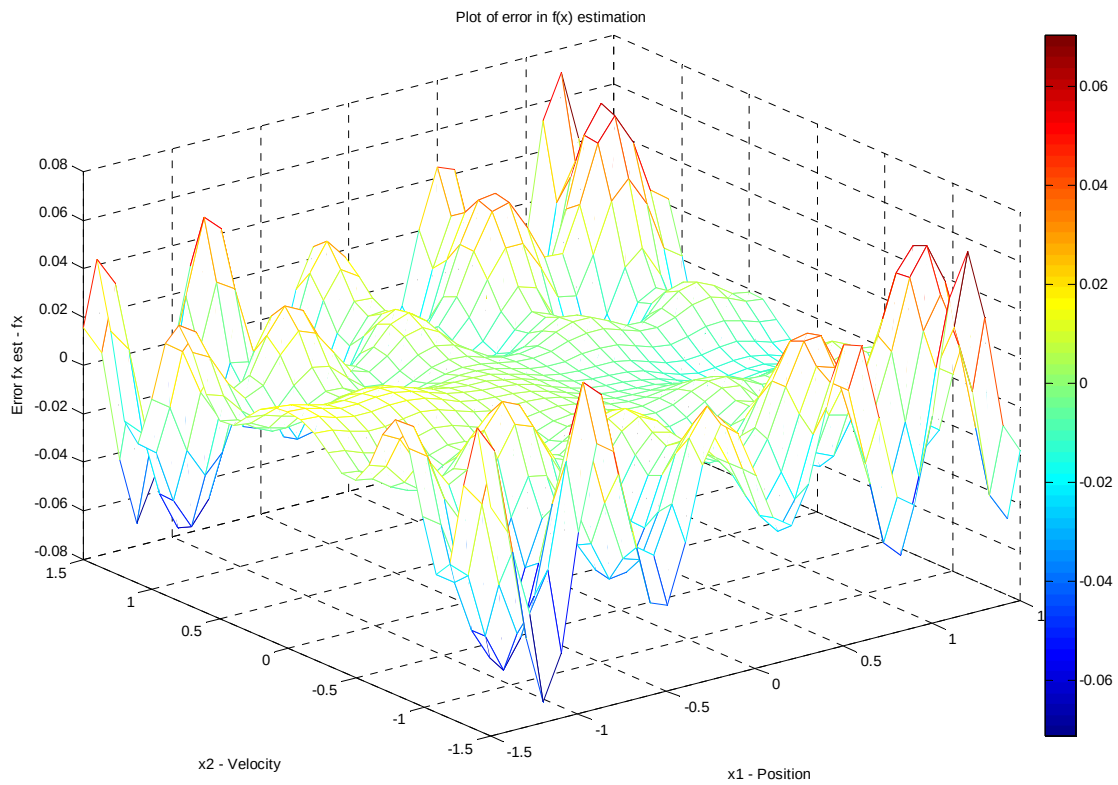


Figure 18. Plot of errors between the function estimated by the identification scheme (selecting 36 nodes) and the function $f(x)$ intended to be estimated

The maximum and average errors calculated are:

Maximum error = $0.0712 = 7.12\%$, Average error = $0.0134 = 1.34\%$

The plot of the estimation of $f(x)$ is also appeared in a separate figure below:

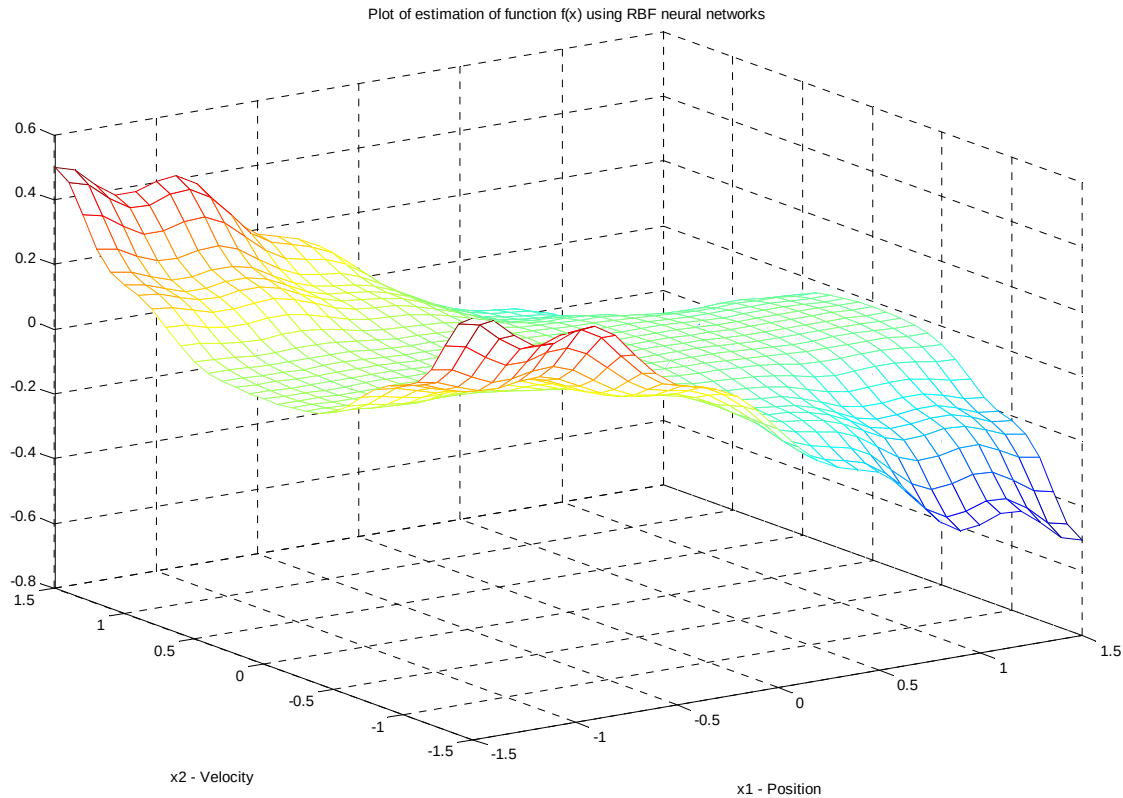


Figure 19. Plot of estimation of function $f(x)$ calculated from the identification scheme selecting 36 nodes inside the compact set

It is concluded, as expected, from the above figures 16,17,18 that the identification of the nonlinear function $f(x)$ is significantly improved giving better estimation of the geometry than in previous simulation of 16 selected nodes while the computation time is also increased.

Simulation with 64 nodes

The identification is improved more compared to the previous case by increasing the selected nodes to 64 inside the compact set. The controller parameters remain the same as in the previous simulation, while the identifier parameters are changed as: $\gamma_f = 2, \sigma_f = 0.0000001$ and the width of RBFs as $\sigma = 0,24$.

The convergence of the weight vector parameters W_f estimation to the optimal W_f^* is presented in the following figure 20:

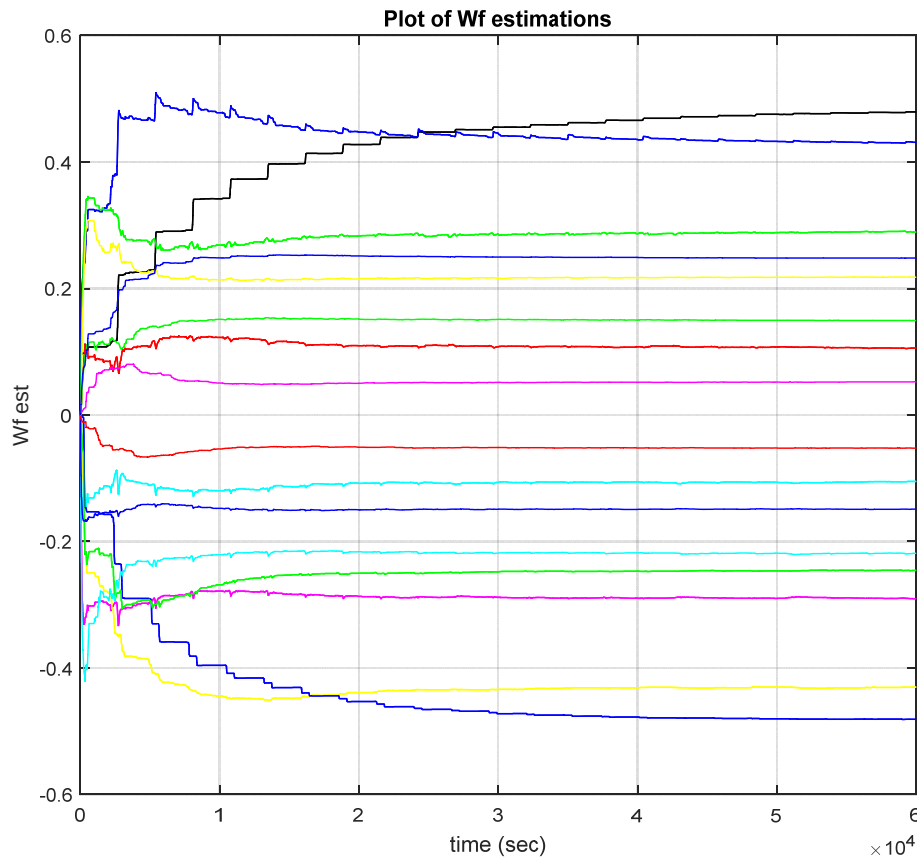


Figure 20. Plot of W_f estimations calculated by the identifier (only the first 16 W_f estimations are appeared in this plot)

The approximation errors of this convergence are:

Maximum estimation error = 0.63% , Average estimation error = 0.11%

The figures showing the efficiency of the estimation of the nonlinear function $f(x)$ are following below:

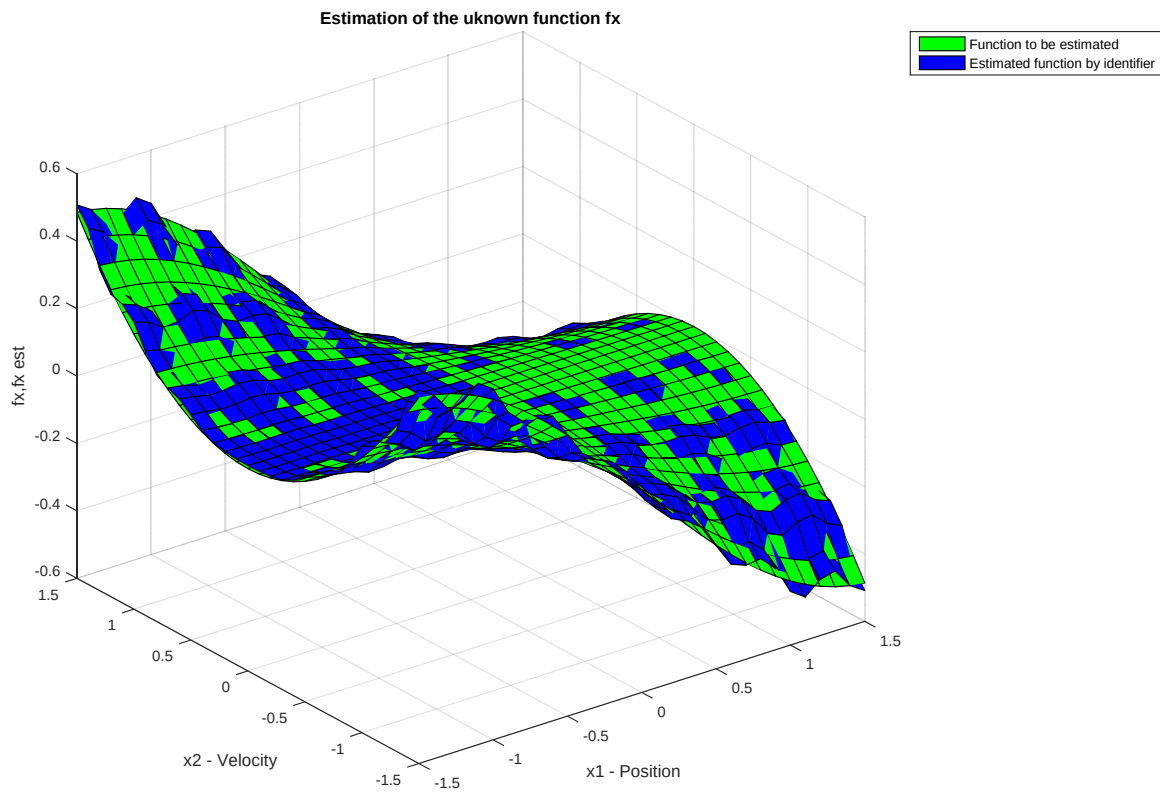


Figure 21. Plot of the function estimated by the identification scheme (selecting 64 nodes) along with the function $f(x)$ intended to be estimated

In the following plot the errors between the estimated non-linear function and the real nonlinear function are appeared giving an evaluation of the learning efficiency.

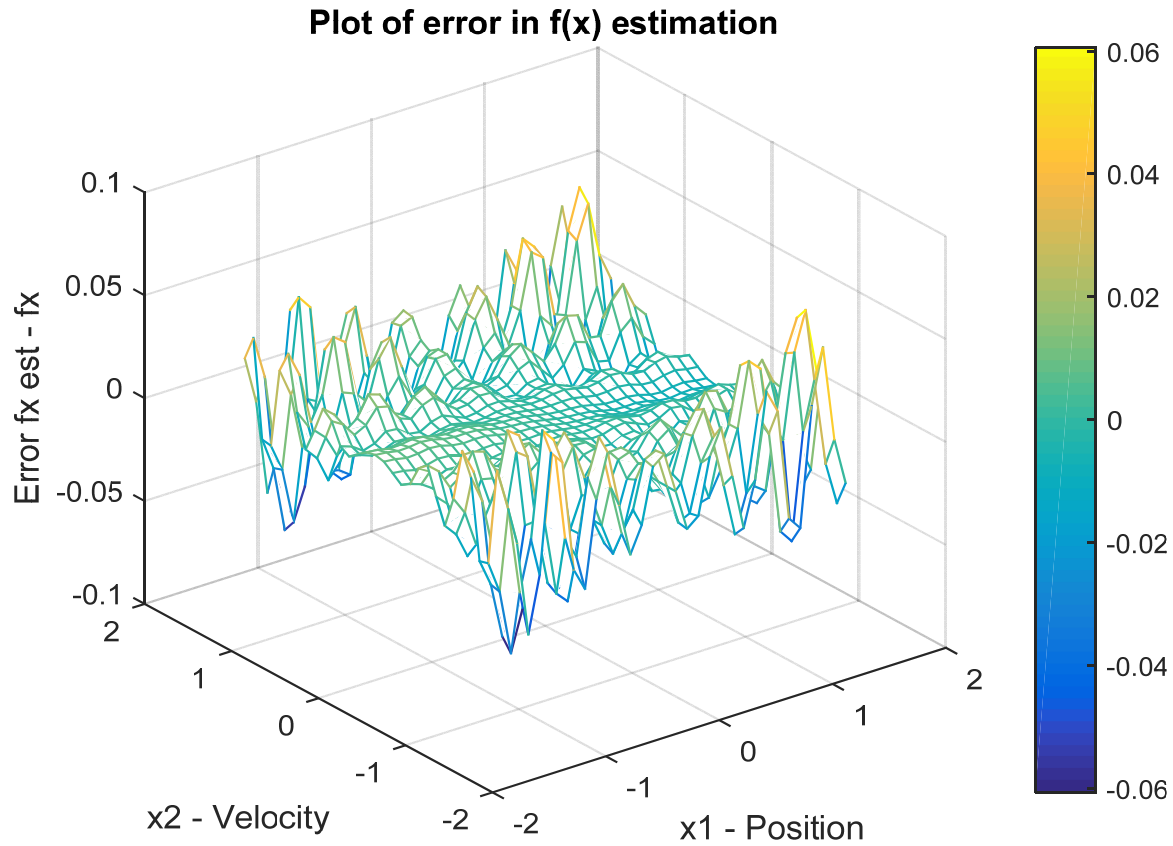


Figure 22. Plot of errors between the function estimated by the identification scheme (selecting 64 nodes) and the function $f(x)$ intended to be estimated

The maximum and average errors calculated are:

Maximum error = $0.0607 = 6.07\%$, Average error = $0.0102 = 1.02\%$

The plot of the estimation of $f(x)$ is also appeared in a separate figure below:

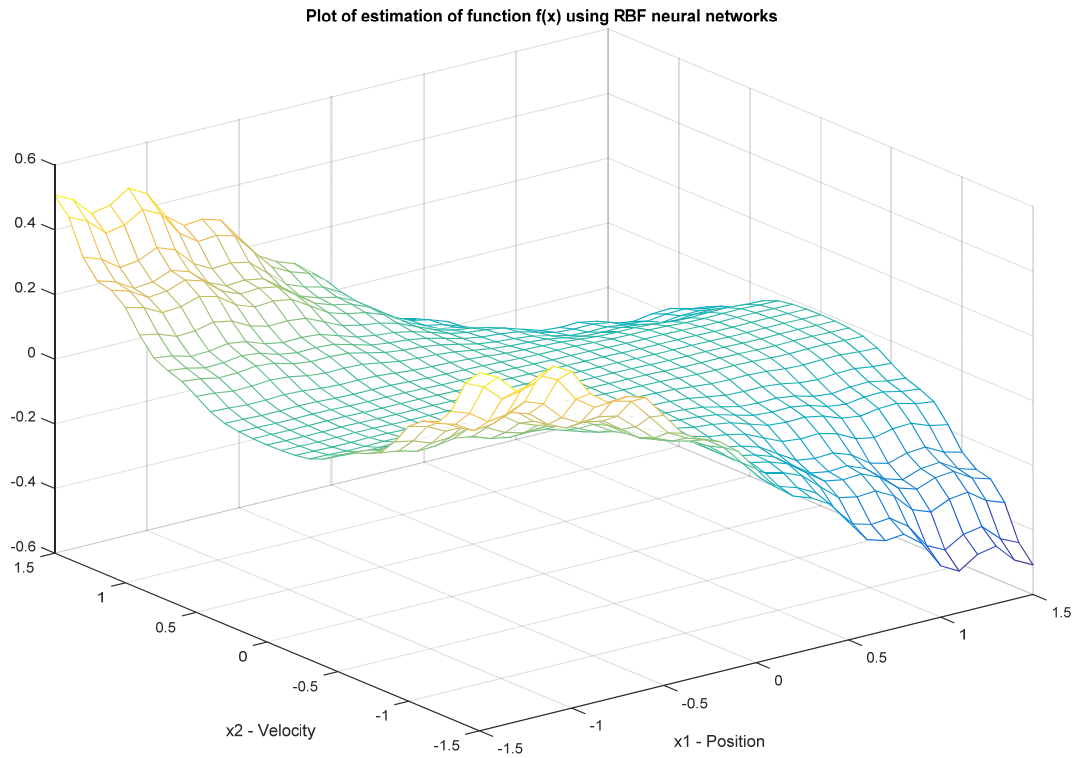


Figure 23. Plot of estimation of function $f(x)$ calculated from the identification scheme selecting 64 nodes inside the compact set

The results of these three simulations appeared above are summarized in the following table:

Nodes of simulation	Maximum error of W_f^* convergence	Average error of W_f^* convergence	Maximum error of $f(x)$ function estimation	Average error of $f(x)$ function estimation	Simulation time [sec]	Computational time [h]
16	0.12%	0.02957%	9.56%	2.18%	2500	0.16
36	0.61%	0.093321%	7.12%	1.34%	18000	0.5
64	0.63%	0.11%	6.07%	1.02%	60000	6.5

Table 1. Results showing the efficiency of learning after three simulations of 16,36,64 nodes

It is proven from the above table that, as the number of selected nodes inside the compact set is increased the result of learning is significantly improved. This is also arisen from the fact that the maximum and average error of $f(x)$ function estimation is reduced as the nodes are increased. Additionally, the increase of the percentage of maximum and average errors of actual values of W_f^* approximation is justified by the fact that as the nodes are increased, then the number of W_f^* quantities (which are equal with the number of nodes) are also increased and the identifier is attempting to succeed the convergence of an increasing number of W_f^* quantities. So in the first simulation the identifier is attempting to manage a convergence of 16 W_f^* quantities while in the third simulation 64 W_f^* quantities correspondingly and the maximum and average error of W_f^* convergence is also minimally increased. It is also important to be emphasized that as the nodes are increased more simulation time is needed by the identifier in order to achieve convergence of W_f^* optimal values, while the computation time of computer vertically increased.

Example 2 – Second order Lagrange system of one degree of freedom with nonlinear $f(x)$ and $g(x)$

A second order Lagrange system of one degree of freedom is proposed which takes the following form:

$$\frac{1}{(\cos q + 0.2)(\dot{q} + 0.15)^2} \ddot{q} + \frac{ml \sin q (\dot{q})^2}{(\cos q + 0.2)(\dot{q} + 0.15)^2} = u \quad (4.9)$$

As the Lagrangian systems has the general form $B(q)\ddot{q} + C(\dot{q}, q) + G(q) = u$ (4.10), it is:

$$B(q) = \frac{1}{(\cos q + 0.2)(\dot{q} + 0.15)^2}$$

$$C(\dot{q}, q) = \frac{ml \sin q (\dot{q})^2}{(\cos q + 0.2)(\dot{q} + 0.15)^2}$$

$$G(q) = 0$$

Also m, l are system's constant parameters denoted as:

m ...mass

l ...length

It has been chosen such a Lagrange system which when it is transformed into canonical form both $f(x), g(x)$ functions are unknown variable nonlinear functions which are intended to be identified in a local region of values applying the learning technique analyzed in this thesis. If we put x_1 as the position and x_2 the velocity of the system, then we have:

$$\begin{aligned}x_1 &= q \\x_2 &= \dot{q}\end{aligned}$$

So, the above Lagrangian equation (4.9) can be transformed to:

$$\ddot{q} = -ml \sin q (\dot{q})^2 + (\cos q + 0.2)(\dot{q} + 0.15)^2 u \quad (4.11)$$

Using the equations $x_1 = q$ and $x_2 = \dot{q}$ the system in canonical form is:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -ml \sin x_1 (x_2)^2 + (\cos x_1 + 0.2)(x_2 + 0.15)^2 u\end{aligned}$$

So, according to the general equation form of (2.62) it holds that:

$$\begin{aligned}f(x) &= -ml \sin x_1 (x_2)^2 \\ g(x) &= (\cos x_1 + 0.2)(x_2 + 0.15)^2\end{aligned}$$

In order to satisfy the assumption 1 (page 35) stated in the problem formulation, in which the equation $g(x)$ which must be positive or negative, then we transform $g(x)$ equation into:

$$g(x) = (\cos x_1 + 0.2)(x_2 + 0.15)^2 + a$$

Where a a positive number such that the $g(x)$ does not take values close to zero. This transformation does not affect the identification result as the $g(x)$ function is moved vertically in z -axis while the variables x_1, x_2 are changing values in x, y -axis.

The parameters chosen in this example are:

$$m = 0.333, \quad l = 0.65$$

So the equations $f(x), g(x)$ which are going to be identified are:

$$f(x_1, x_2) = -0.21645 \sin(x_1) \cdot x_2^2 \quad (4.12)$$

$$g(x_1, x_2) = (\cos x_1 + 0.2)(x_2 + 0.15)^2 + a \quad (4.13)$$

The identification scheme presented in chapter four is used for local identification and in this example the identification is selected to be done in the following area:

$$\begin{aligned} -1.5 &\leq x_1 \leq 1.5 \\ -1.5 &\leq x_2 \leq 1.5 \end{aligned}$$

According to the identification methodology the nonlinear functions $f(x), g(x)$ are approximated by the following linear forms:

$$f(x) = W_f^{*T} Z_f(x) + \varepsilon_f(x) \quad (4.14)$$

$$g(x) = W_g^{*T} Z_g(x) + \varepsilon_g(x) \quad (4.15)$$

$\varepsilon_f(x), \varepsilon_g(x) \dots$ are the error terms

$W_f^{*T}, W_g^{*T} \dots$ are the optimal weight vectors

$Z_f(x), Z_g(x) \dots$ are the regressor vectors

The elements of the regressor vectors $Z_f(x) = [zf_1 \ \dots \ zf_n]^T$ where $Z_f(x) \in \Re^n$ and $Z_g(x) = [zg_1 \ \dots \ zg_n]^T$ where $Z_g(x) \in \Re^n$ are calculated from the following equations:

$$zf_i(x) = \exp\left(-\left\|\frac{x - cf_i}{\sigma_1}\right\|^2\right) \quad (4.16), \ i = 1, \dots, n$$

$$zg_i(x) = \exp\left(-\left\|\frac{x - cg_i}{\sigma_2}\right\|^2\right) \quad (4.17), \ i = 1, \dots, n$$

Where n is the number of nodes placed in the compact set.

$cf_i, cg_i \dots$ are equal vectors of the coordinates of each of the selected nodes inside the compact set

Also in the above calculation we have to select both the RBFs widths σ_1, σ_2 . We keep the same selections for the controller parameters. So:

$$\epsilon = 0.5 \min_{i \neq j} \left\{ \frac{1}{2} \|cf_i - cf_j\| \right\} = 0.25$$

$$\lambda = 1.5, k = 12.5$$

$$\lim_{t \rightarrow \infty} \rho_{\varepsilon}(t) = 0.5 \frac{\epsilon}{2 \sqrt{\sum_{i=1}^n \left(\frac{2^{i-1}}{\lambda^{n-i}} \right)^2}} = 0.0419$$

$$\rho_{\varepsilon}(0) = 2 * |\varepsilon(0)| + 0.1 = 2.3757$$

By the above selections of controller parameters, it is guaranteed that the vectors $Z_f(x(t)), Z_g(x(t))$ are persistently exciting and the performance function is:

$$\rho_{\varepsilon}(t) = (\rho_{\varepsilon}(0) - \rho_{\infty}(t)) e^{-t} + \rho_{\infty}(t) = 2.3338 e^{-t} + 0.0419$$

Simulation with N=9 nodes

For this example a simulation choosing 9 nodes inside the predefined compact set is made. The desired trajectory of position and velocity are plotted below as a function of time along with the system states showing that the controller action guarantees that system states follow successfully the desired trajectory.

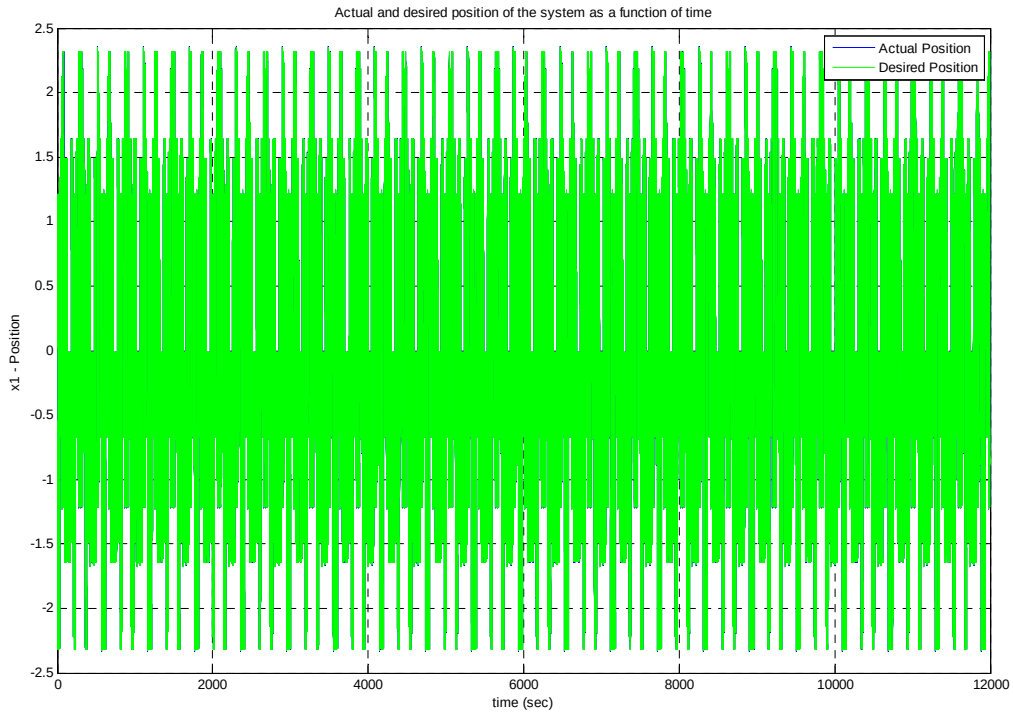


Figure 24. Actual and desired values of x1-position of the system as a function of time

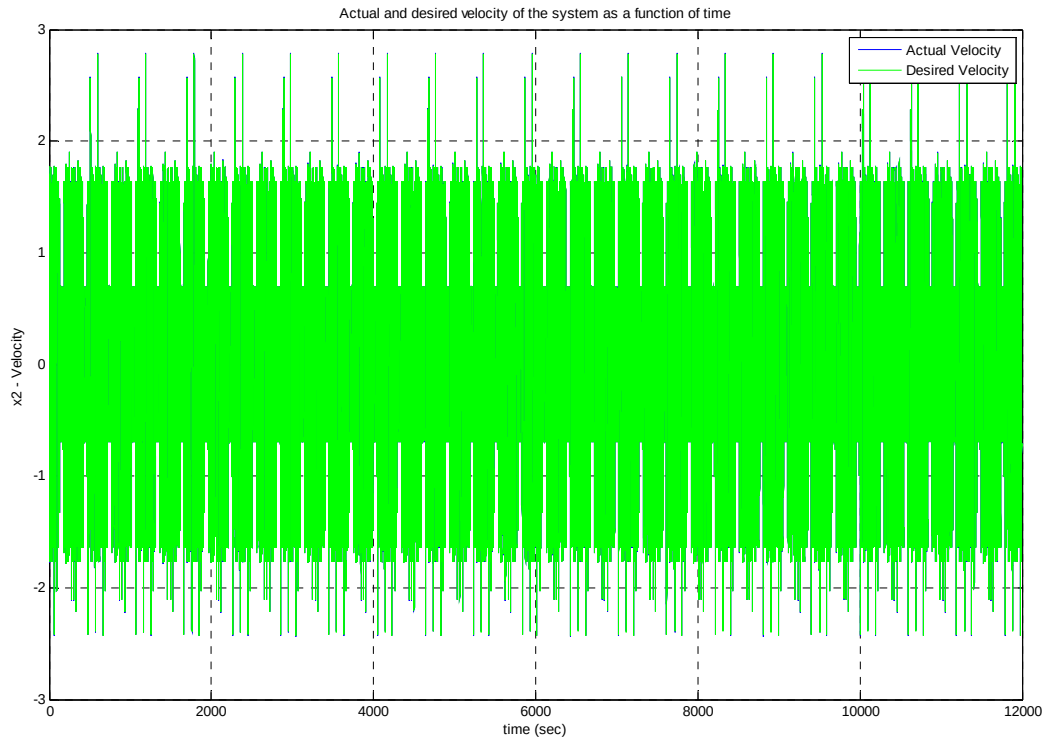


Figure 25. Actual and desired values of x_2 -velocity of the system as a function of time

From the above figures 24,25 it is clear that, the controller guarantees that system states $x_1(t), x_2(t)$ follow the desired trajectory with insignificant error.

The values of control action are also appeared as a function of time in the following figure.

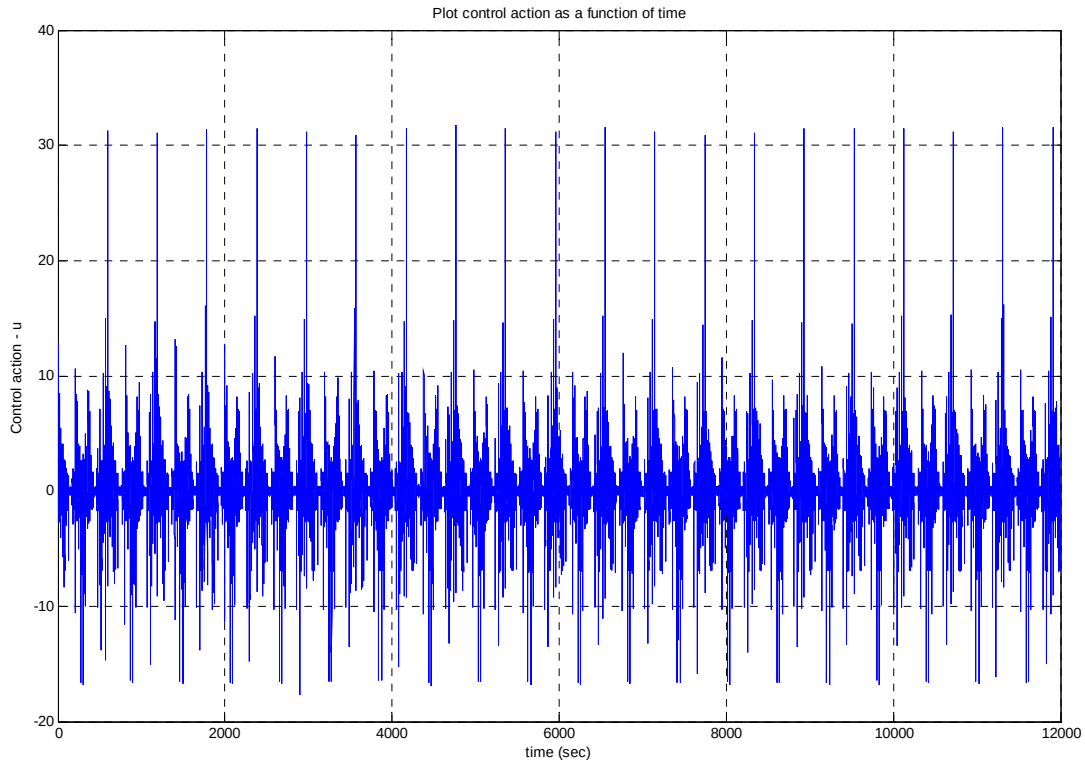


Figure 26. Control action which guarantees that system states follow the desired trajectory

By utilizing the proposed control input, the produced orbit visits periodically an ϵ – neighborhood of the Gaussian centers as it is appeared in the figure 7.

The identification scheme (3.102)-(3.104) was simulated by selecting appropriately the parameters $\gamma_f, \gamma_g, \sigma_f, \sigma_g$ and also the matrix K used in the equation (3.102) in a way that the matrix $A - BK^T$ is Hurwitz. We select $K = [4 \ 4]^T$. For the simulation with 9 nodes in this example we select the parameters $\gamma_f = 2, \gamma_g = 2, \sigma_f = 0.0000001, \sigma_g = 0.0000005$ and also we select the width of RBFs as $\sigma_1 = \sigma_2 = 0.6$.

In this simulation of 9 nodes, the optimal weight vectors W_f^*, W_g^* are calculated as:

$$W_f^* = [0.4858 \ 0 \ -0.4858 \ 0 \ 0 \ 0 \ 0.4858 \ 0 \ -0.4858]$$

$$W_g^* = [2.7371 \ 5.2670 \ 2.7371 \ 2.0061 \ 2.0270 \ 2.0061 \ 2.4934 \ 4.1870 \ 2.4934]$$

The convergence of the weight vector parameters W_f, W_g estimations after the end of simulation are presented in the following figures 27,28.

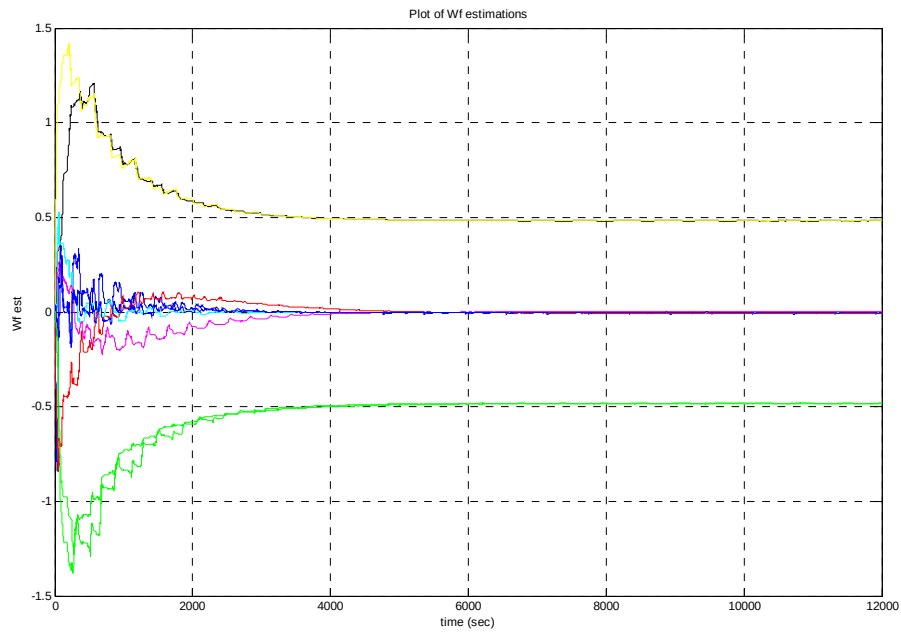


Figure 27. Plot of W_f estimations calculated by the identifier selecting 9 nodes

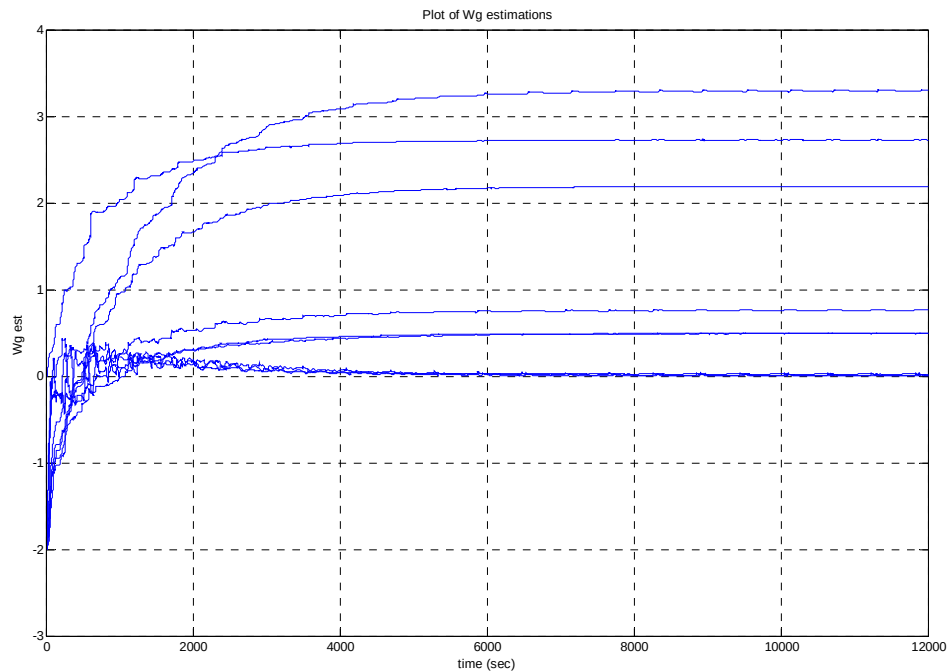


Figure 28. Plot of W_g estimations calculated by the identifier selecting 9 nodes

As it was stated in the theoretical analysis the estimated parameters converge to a small area of their actual (optimal) values achieving the target of learning the unknown nonlinear functions $f(x), g(x)$. The convergence can be evaluated by calculating the maximum and average estimation error between the W_f, W_g estimated values and the actual ones.

It is for W_f estimation:

Maximum estimation error = 0.0079 = 0.79% , Average estimation error = 0.0037 = 0.37%

Also for W_g estimation:

Maximum estimation error = 0.0277 = 2.77% , Average estimation error = 0.0075 = 0.75%

Also the functions produced by the identifier as estimations of the nonlinear functions $f(x), g(x)$ are plotted in the following figures along with the nonlinear functions $f(x), g(x)$ with the intention of evaluating the learning quality of the proposed technique.

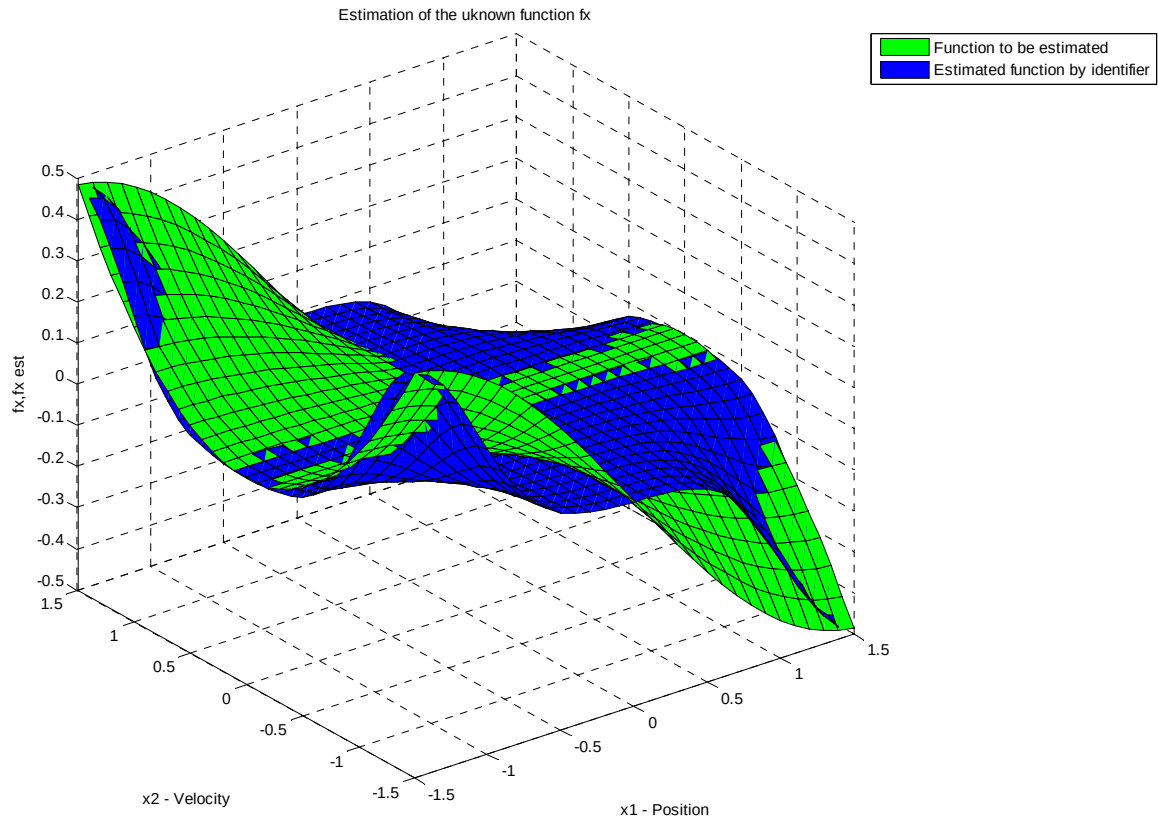


Figure 29. Plot of the function estimated by the identification scheme (selecting 9 nodes) along with the function $f(x)$ intended to be estimated

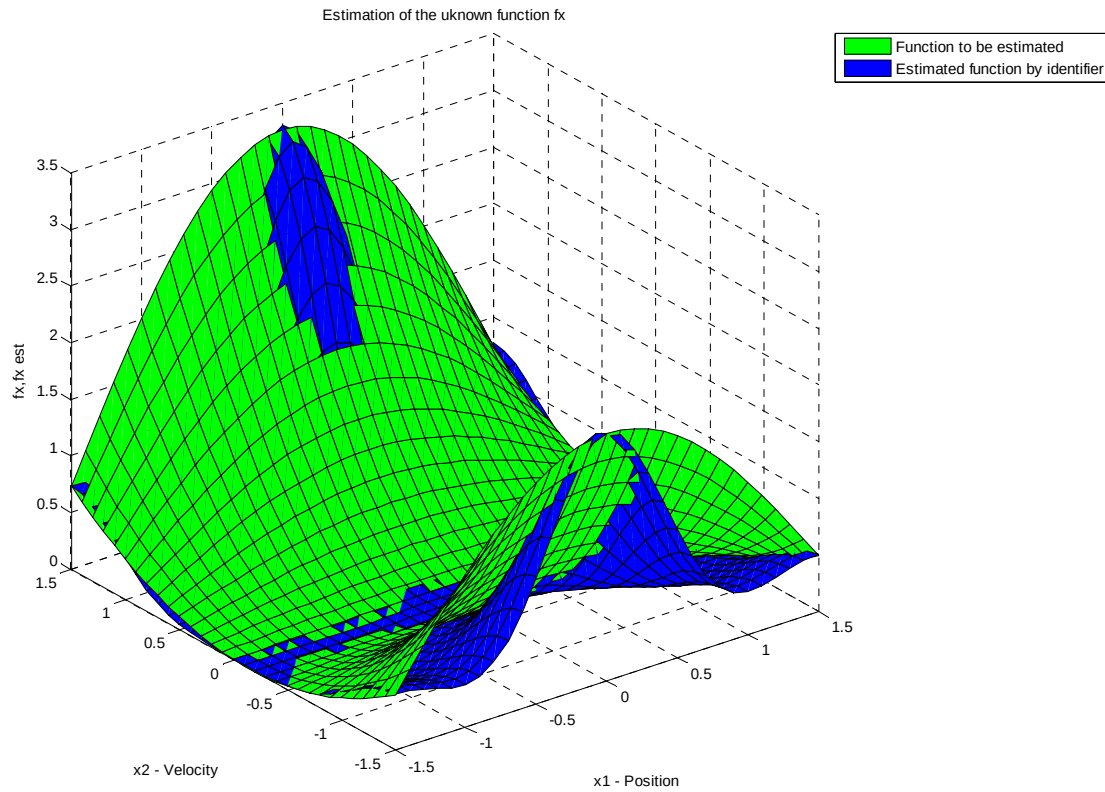


Figure 30. Plot of the function estimated by the identification scheme (selecting 9 nodes) along with the function $g(x)$ intended to be estimated

Two more plots are also appeared showing the errors between the real nonlinear function and the estimation of this function.

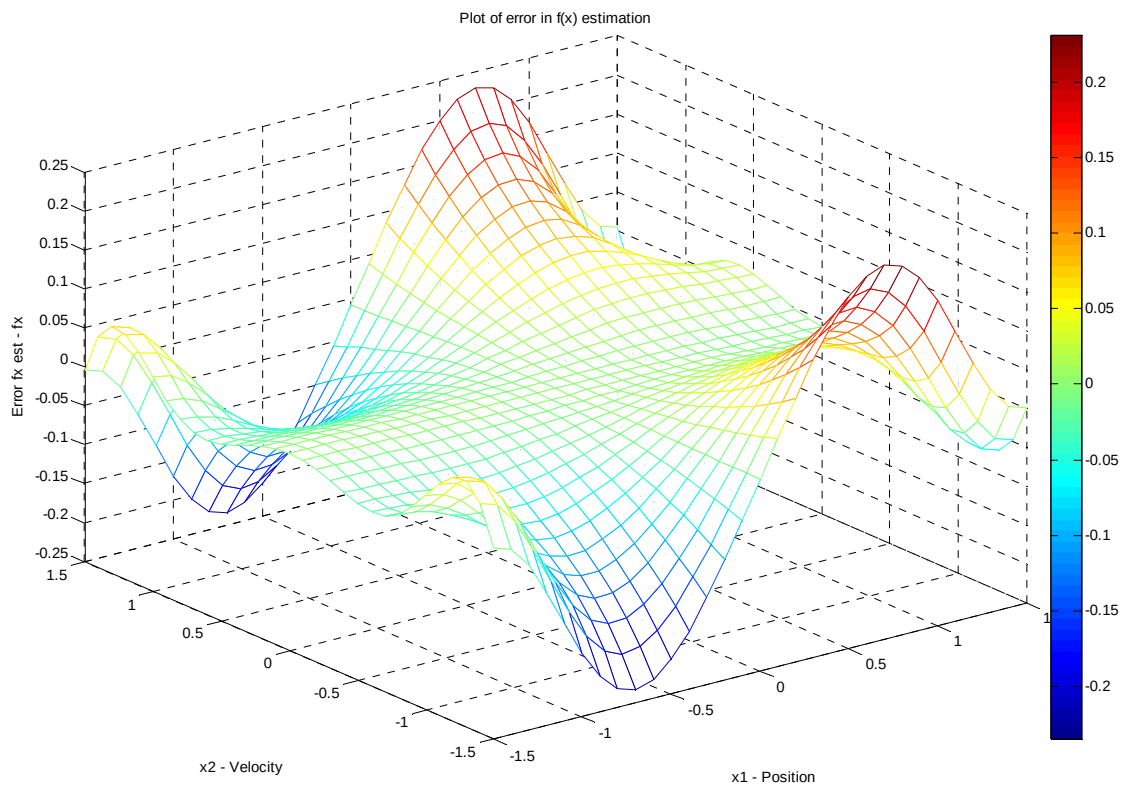


Figure 31. Plot of errors between the function estimated by the identification scheme (selecting 9 nodes) and the function $f(x)$ intended to be estimated

The maximum and average errors calculated are:

Maximum error = $0.2342 = 23.42\%$, Average error = $0.0462 = 4.62\%$

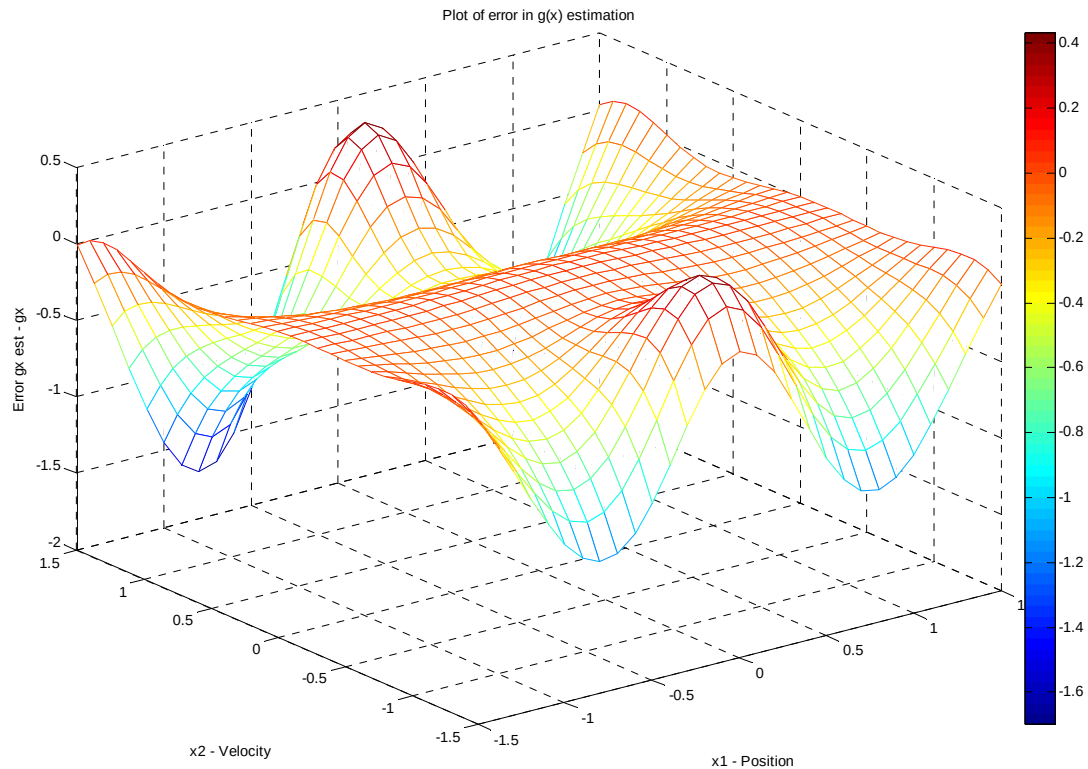


Figure 32. Plot of errors between the function estimated by the identification scheme (selecting 9 nodes) and the function $g(x)$ intended to be estimated

The maximum and average errors calculated are:

Maximum error = $1.6969 = 169.69\%$, Average error = $0.2813 = 28.13\%$

The plot of the estimated functions $f(x), g(x)$ are also appeared in separate figures below:

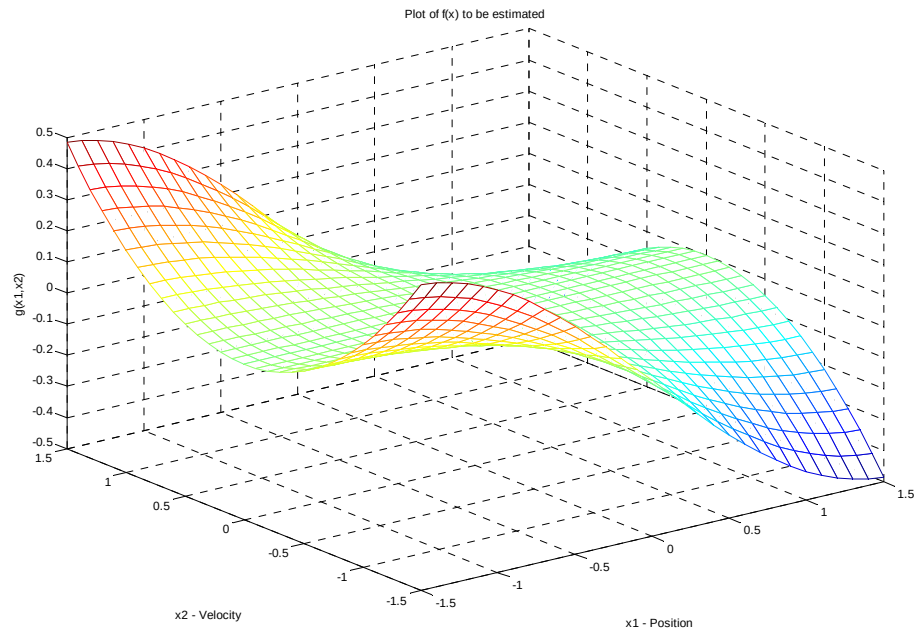


Figure 33. Plot of function $f(x)$ to be estimated by the identification scheme

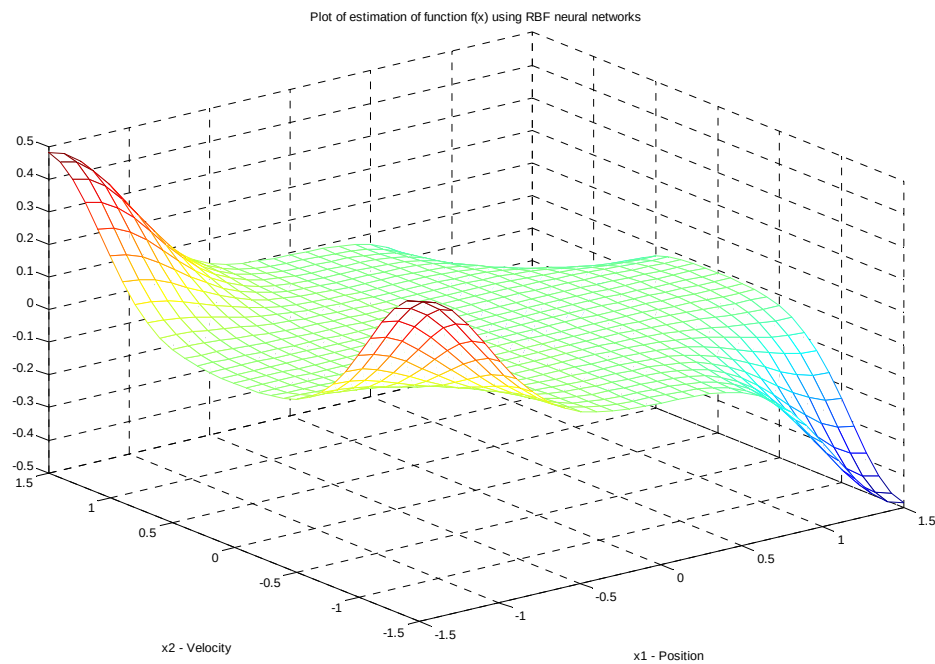


Figure 34. Plot of estimation of function $f(x)$ calculated from the identification scheme selecting 9 nodes inside the compact set

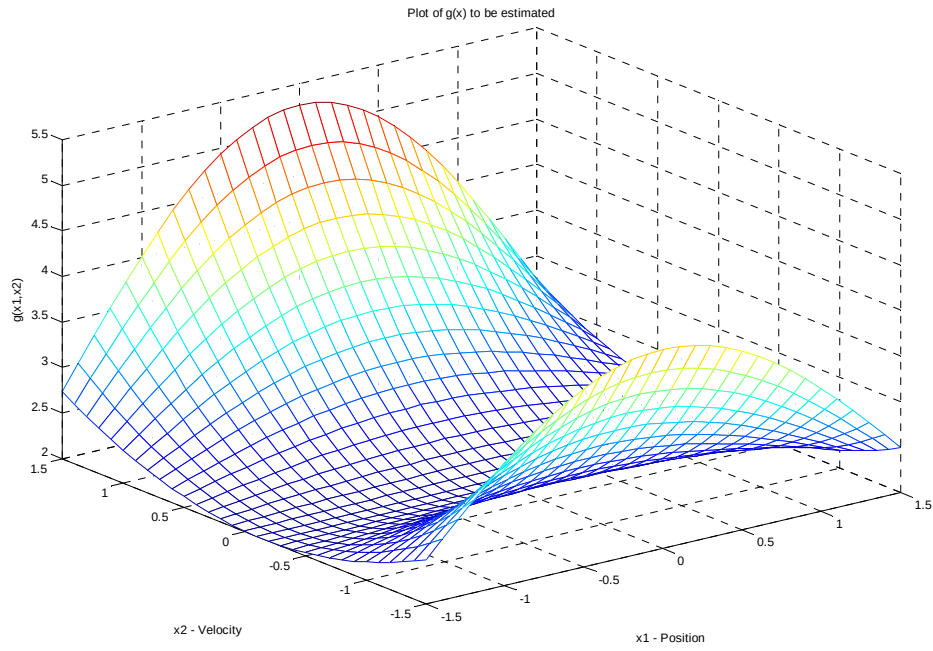


Figure 35. Plot of function $g(x)$ to be estimated by the identification scheme

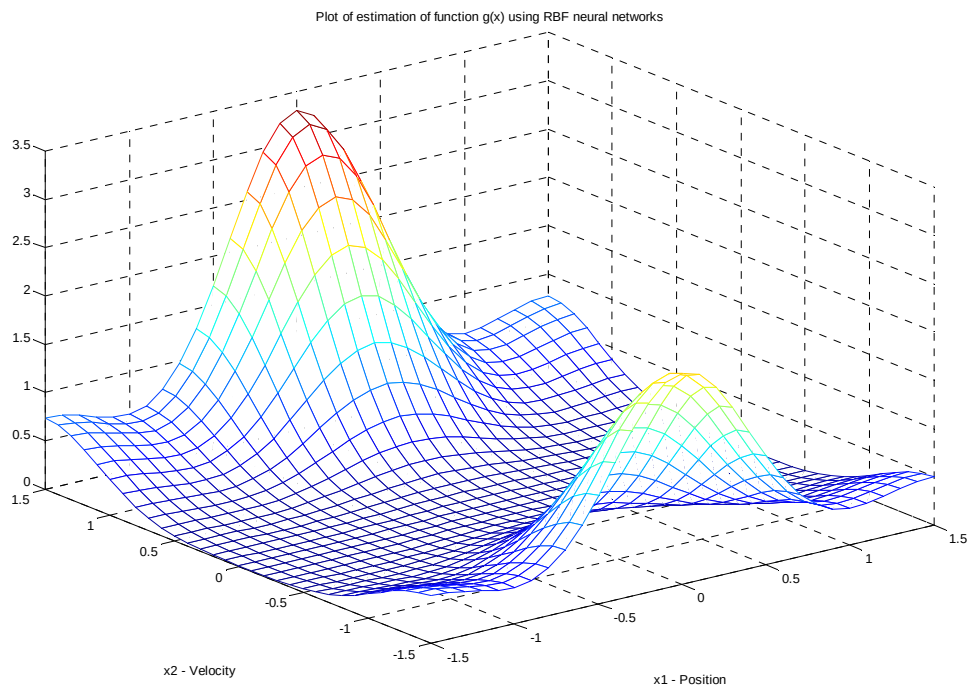


Figure 36. Plot of estimation of function $g(x)$ calculated from the identification scheme selecting 9 nodes inside the compact set

From the above figures 30-35 it is concluded that, the target of learning is achieved producing a local satisfactory approximation of the unknown nonlinear functions $f(x), g(x)$ in the area of interest where the compact set is defined. The approximation errors shown in the figures 31,32 can be reduced if the number of the selected nodes inside the compact set is increased. By increasing the number of nodes, more system state values will be stimulated by the learning technique which will be identified by the identifier, having as a drawback the increase of the computation time. (In the next simulation we increase the number of selected nodes in order to enhance the results of learning).

Simulation with N=16 nodes

In this simulation the number of the selected nodes is increased to 16. Also we select the parameters $\gamma_f=2, \gamma_g=2, \sigma_f=0.0000001, \sigma_g=0.0000005$ and the width of RBFs as $\sigma_1=0.6, \sigma_2=0.55$.

The convergence of the weight vector parameters W_f, W_g estimations after the end of simulation are presented in the following figures 37,38.

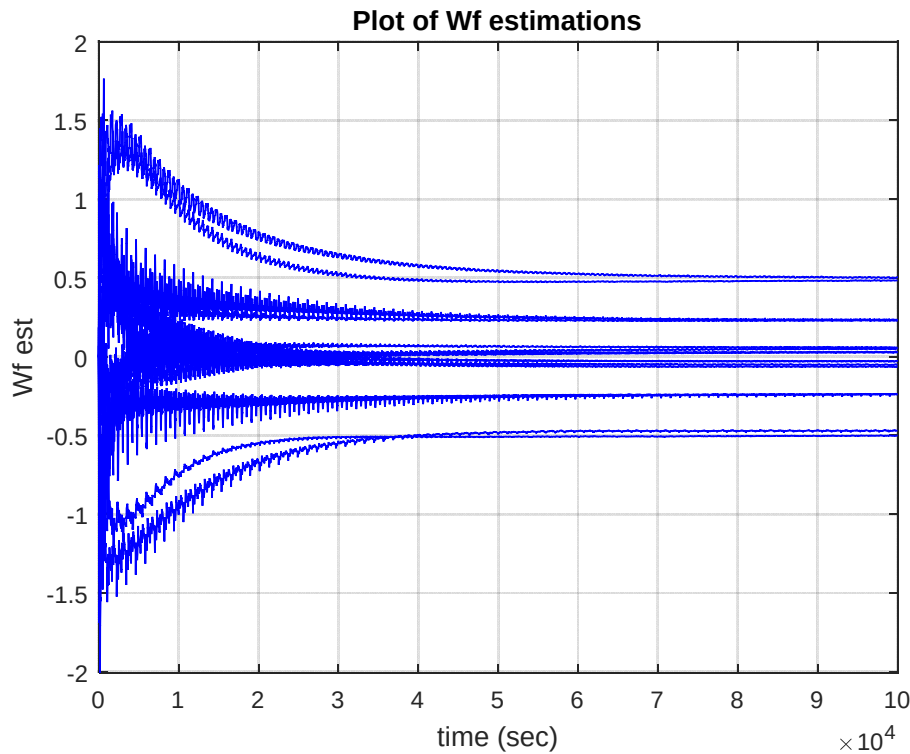


Figure 37. Plot of W_f estimations calculated by the identifier selecting 16 nodes

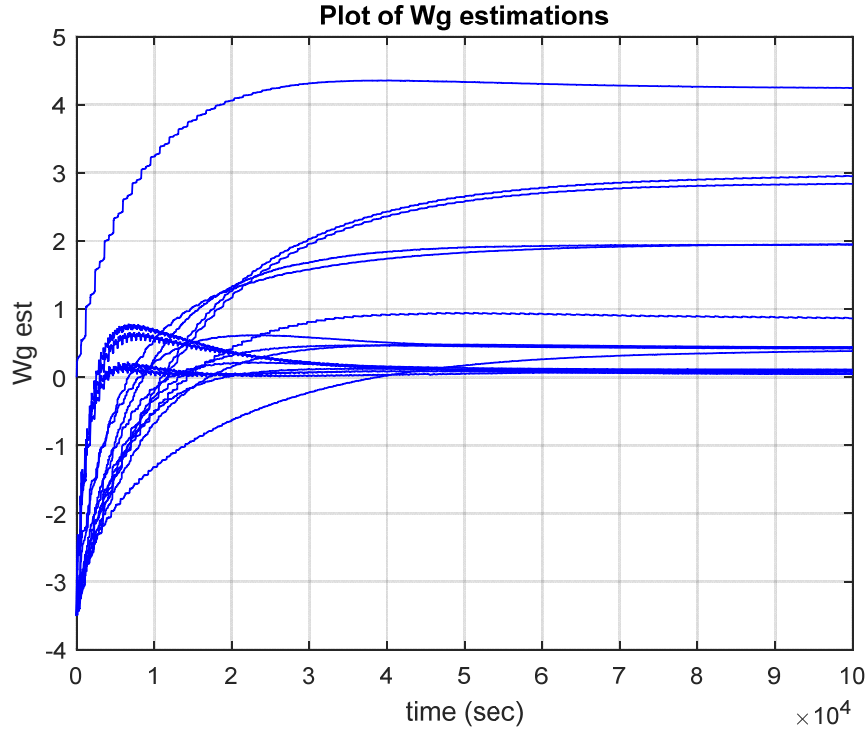


Figure 38. Plot of W_g estimations calculated by the identifier selecting 16 nodes

The convergence can be evaluated by calculating the maximum and average estimation error between the W_f, W_g estimated values and the actual ones.

It is for W_f estimation:

Maximum estimation error = 0.0189 = 1.89% , Average estimation error = 0.0057 = 0.57%

Also for W_g estimation:

Maximum estimation error = 0.1251 = 12.51% , Average estimation error = 0.0396 = 3.96%

Also the functions produced by the identifier as estimations of the nonlinear functions $f(x), g(x)$ are plotted in the following figures along with the nonlinear functions $f(x), g(x)$ with the intention of evaluating the learning quality of the proposed technique.

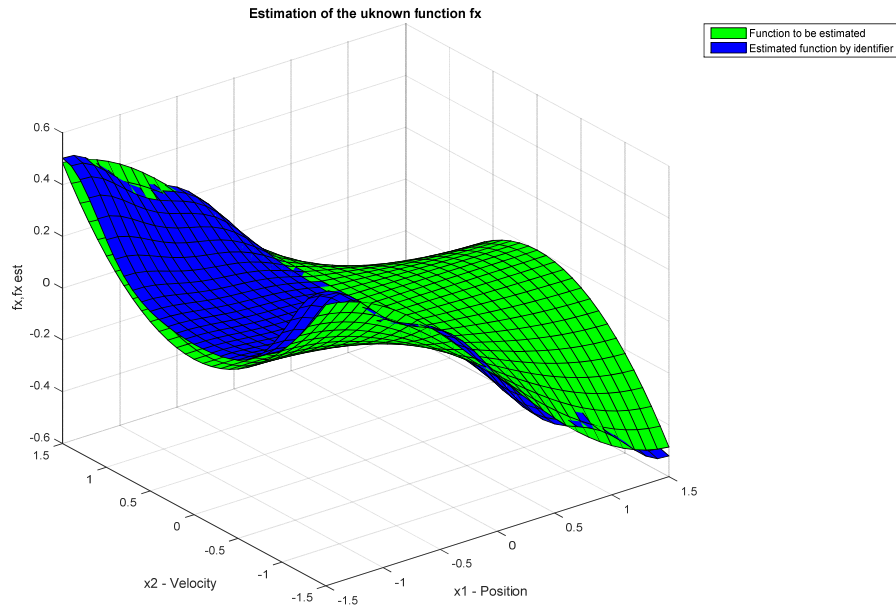


Figure 39. Plot of the function estimated by the identification scheme (selecting 16 nodes) along with the function $f(x)$ intended to be estimated

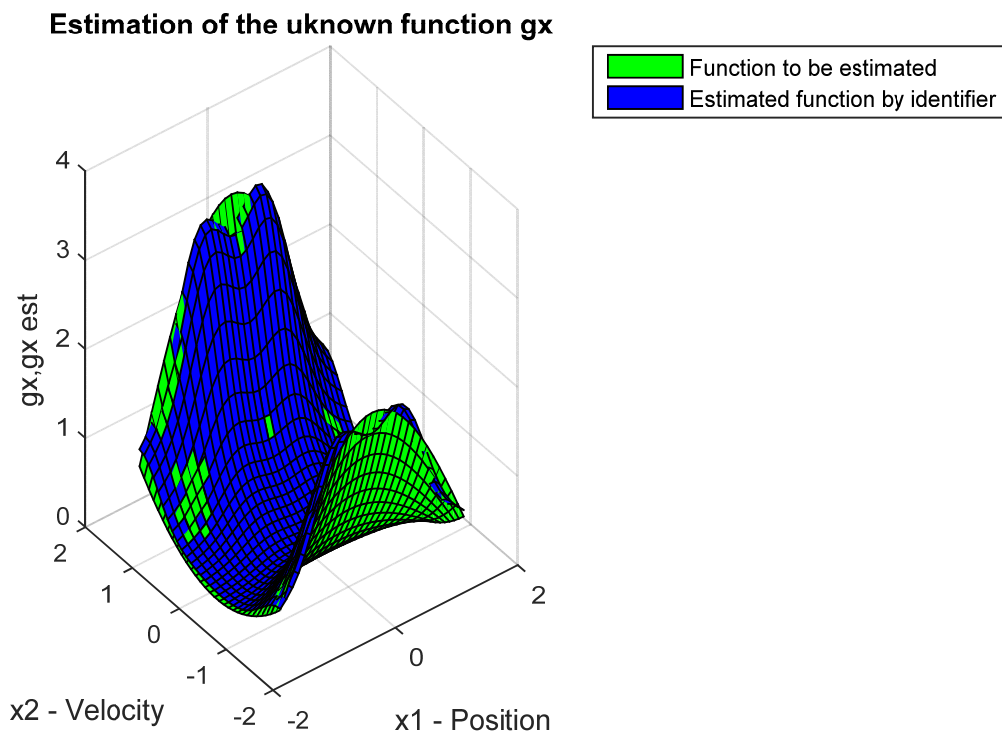


Figure 40. Plot of the function estimated by the identification scheme (selecting 16 nodes) along with the function $g(x)$ intended to be estimated

Two more plots are also appeared showing the errors between the real nonlinear function and the estimation of this function.

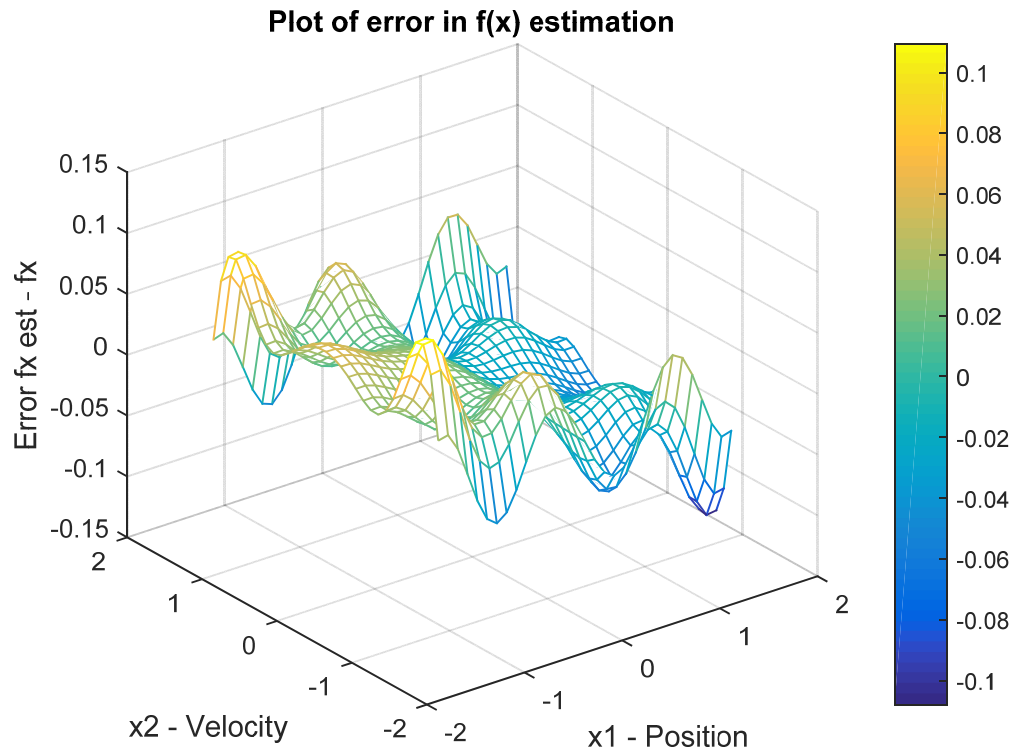


Figure 41. Plot of errors between the function estimated by the identification scheme (selecting 16 nodes) and the function $f(x)$ intended to be estimate

The maximum and average errors calculated are:

Maximum error = 0.1094 = 10.94% , Average error = 0.0310 = 3.1%

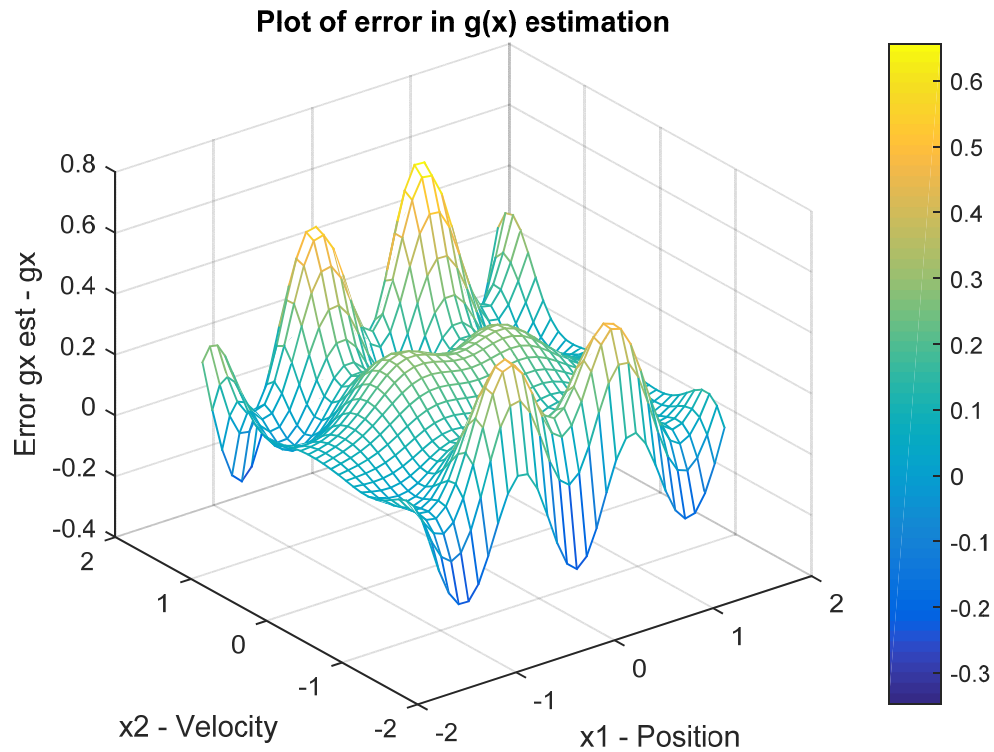


Figure 42. Plot of errors between the function estimated by the identification scheme (selecting 16 nodes) and the function $g(x)$ intended to be estimated

The maximum and average errors calculated are:

Maximum error = $0.6563 = 65.63\%$, Average error = $0.1645 = 16.45\%$

The basic conclusion in this simulation is that as the number of nodes inside the compact set is increased, the maximum and average errors in $f(x), g(x)$ estimations are decreased and the geometry of the nonlinear surface of both functions is better approximated as it is shown in the figures 39-40.

CHAPTER 5 – CONCLUSIONS & FUTURE WORK

Conclusions

In this thesis a system identification methodology has been designed for on-line learning of the actual nonlinearities of second order Lagrangian systems with n degrees of freedom. The whole scheme is consisted of an online RBF neural network identifier and a controller achieving two goals. Firstly it guarantees that system states follow with insignificant error a desired trajectory inserted to the system. Also it generates such control signals which satisfy the persistency of excitation condition for the RBF regressors employed in the identification scheme. The nonlinear characteristics of the system are expressed by neural networks in linear form as a product between the estimated weight vector and the regressor vector. It has been proven in this work, that if the persistency of excitation condition is satisfied, then the estimated weight vector converges to the optimal weight vector with minimal error and the result or learning is achieved. The result of learning for the nonlinearities of the system is achieved locally in a predefined compact set whose coordinates are the two states of the system, position state as the x - axis and velocity state as the y - axis. So this compact set includes all possible states of the system inside this region of values. The desired trajectory designed in this methodology plays an important role contributing to the achievement of learning. Initially inside the predefined set we select specific number of nodes where the centers of RBFs are put. The desired trajectory is smooth traversing all the centers of the selected RBFs periodically. This fact satisfies the persistency of excitation condition. The desired trajectory is constituted from a set of separate ellipses and each ellipse is designed with a certain technique ensuring that it passes at least through one of the nodes of the compact set. So there are so many ellipses as the total number of the selected nodes. If the number of the selected nodes is increased then, the states of the system stimulated by the desired trajectory is also increased improving the final result of learning increasing in parallel the computation load.

There are two identical characteristics of this approach. The first one is the fact that an autonomous system is proposed which produces a desired trajectory able to stimulate all possible state values of the system to be identified in a local region of values and the other one is the isolation between the identifier and the controller design increasing the robustness level of the learning scheme. Finally the theoretical findings are explained and verified by different simulation examples of Lagrangian systems whose non-linear parts are identified and the effectiveness of the method is evaluated through the figures shown in each example.

The advantages of this learning technique can be used in many applications. As almost all mechanical and electrical systems follow Lagrange dynamics, then systems of this category whose model is complex, unknown and highly nonlinear can be identified with minimal

convergence error. For example vehicles models (like submarines) can be learned in a controlled space (predefined compact set) and then state estimators (observers) can be developed for them giving good estimations in environments under unknown conditions. Also for systems whose modeling is known, then various kinds of controllers can be designed as well as its tuning process can be facilitated achieving the desired control performance. This learning mechanism can also be useful in robotics field.

Future work

There are many kinds of extensions for future work based in this learning mechanism developed in this thesis. By acquiring a complete model of an unknown system then a future work could be to design an observer (like a Kalman filter) giving accurate state estimations of this system. Also as this work is based in Lagrangian systems which can be transformed into canonical form, it can also be extended for other kind of systems like pure-feedback systems and strict-feedback systems. Another interesting field is the fault detection and diagnosis which plays important role in modern engineering systems. As stated in paper [49] there is an increasing demand for high reliability and safety in many complex industrial processes. In recent years, for nonlinear systems with structured modeling uncertainties and nonlinear faults, in which the faults cannot be decoupled from unknown inputs, different diagnosis methods have been proposed. To detect the unknown faults, accurate models of the monitored system underlying normal and fault modes are required. Neural networks can provide powerful tools for this task. So, a kind of future work could be the fault detection of a certain kind of systems (like strict-feedback systems) using the learning methodology proposed in this thesis.

REFERENCES

- [1] Panos J. Antsaklis, “*Intelligent Learning Control*”, Guest Editor’s Introduction, IEEE Control Systems Vol.15, No.3, pp.5-7, June 1995
- [2] Jay Farrell, Walter Baker, book “*An introduction to intelligent and autonomous control*” Vol.10 Learning control systems pages 237-262, Kluwer Academic Publishers Norwell, MA, USA 1993
- [3] Cong Wang, David J. Hill, “*Deterministic learning theory for identification, recognition, and control*”, CRC Press 2010
- [4] Hiroyuki Miyamoto, Mitsuo Kawato, Tohru Setoyama and Ryoji Suzuki, “*Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator*”, Neural Networks Vol.1, pp. 251-265, 1988
- [5] Ronald M. Holdaway, Mark W. White “*Computational Neural Networks: Enhancing supervised learning algorithms via self-organization*”, Int J Biomed Comput 25 (1990)
- [6] Nico Waymaere, Jean-Pierre Martens, “*A Fast and Robust Learning Algorithm for Feedforward Neural Networks*”, Neural Networks, Vol.4, pp. 361-369, 1991
- [7] J Yuh, “*A Multilayered Neural Net Controller Using Direct Learning Algorithm*”, Computers Elect. Engng, Vol.19, No. 4, pp. 225-264, 1993
- [8] Charles W. Lee, “*Learning in Neural Networks by Using Tangent Planes to Constraint Surfaces*”, Neural Networks, Vol. 6, pp. 385-392, 1993
- [9] C.Lee Giles, B.G. Horne, T.Lin, “*Learning a Class of Large Finite State Machines with a Recurrent Neural Network*”, Neural Networks, Vol. 8, No. 9, pp. 1359-1365, 1995
- [10] C. James Li “*Mechanical system modeling using recurrent neural networks via quasi-Newton learning methods*” Appl. Math. Modeling, 1995, Vol. 19, July
- [11] Songwu Lu., Tamer Basar, “*Robust Nonlinear System Identification Using Neural-Network Models*”, IEE Transactions on Neural Networks, vol. 9, No.3, May 1998
- [12] Cong Wang, David J. Hill, Guanrong Chen, “*Deterministic Learning of Nonlinear Dynamical Systems*”, Proceedings of the 2003 IEEE, International Symposium on Intelligent Control Houston, Texas, October, 2003
- [13] Cong Wang, David J. Hill, Guanrong Chen, “*Neural Learning Control*”, Proceedings of the 5th World Congress on Intelligent Control and Automation, June 15-19, 2004, Hangzhou, China

- [14] Cong Wang, David J. Hill, “*Learning from direct adaptive neural control*”, Control Conference, 2004. 5th Asian, July 2004
- [15] Cong Wang, David J. Hill, “*Learning from Neural Control*”, IEEE Transactions on Neural Networks, Vol. 17, No. 1, January 2006
- [16] Tengei Liu, Cong Wang, “*Learning from Neural Control of General Brunovsky Systems*”, Proceedings of the 2006 IEEE, International Symposium on Intelligent Control, Munich, October 4-6, 2006
- [17] Tengei Liu, Cong Wang, “*Learning from Neural Control of Strict-feedback Systems*”, Proceedings of the 2007 IEEE International Conference on Control and Automation Guangzhou, China – May 30 to June 1, 2007
- [18] Tengei Liu, Cong Wang, “*Learning from Neural Control in Motor Systems*”, Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics December 15-18, 2007, Sanya, China
- [19] Tengei Liu, Cong Wang, David J. Hill, “*Learning from neural control of nonlinear systems in normal form*”.
- [20] Chengzhi Yuan, Cong Wang, “*Persistency of excitation and performance of deterministic learning*”, School of Automation and Center for Control and Optimization, South China University of Technology, Guangzhou 510641, China
- [21] Shi-Lu Dai, Cong Wang, Fei Luo, “*Learning Control of Uncertain Ocean Surface Ship Dynamics Using Neural Networks*”.
- [22] Shi-Lu Dai, Min Wang, Cong Wang, Liejun Li, “*Learning from Stable Adaptive NN Output Feedback Control of Uncertain Ship Dynamics*”, Proceedings of the 31st Chinese Control Conference July 25-27, 2012, Hefei, China
- [23] Cong Wang, Min Wang, Tengfei Liu, David J. Hill, “*Learning From ISS-Modular Adaptive NN Control of Nonlinear Strict-Feedback Systems*”, IEEE Transactions on Neural Networks and Learning Systems, Vol.23, No. 10, October 2012
- [24] Shi-Lu Dai, Cong Wang, Fei Luo, “*Identification and Learning Control of Ocean Surface Ship Using Neural Networks*”, IEEE Transactions on Industrial Informatics, Vol.8, No.4, November 2012
- [25] Shi-Lu Dai, Min Wang, Cong Wang, Liejun Li, “*Learning from adaptive neural network output feedback control of uncertain ocean surface ship dynamics*”, International Journal of Adaptive control and Signal Processing, Int. J. Adapt. Control Signal Process, 2012

- [26] Min Wang, Cong Wang, “*Learning from Adaptive Neural Control for a Class of Pure-Feedback Systems*”, 2012 12th International Conference of Control Automation, Robotics & Vision Guangzhou China, 5-7th December 2012
- [27] Min Wang, Cong Wang, “*Identification and Learning Control of Strict-Feedback Systems Using Adaptive Neural Dynamic Surface Control*”, 2013 IEEE
- [28] Shi-Lu Dai, Cong Wang, Min Wang, “*Dynamic Learning from Adaptive Neural Network Control of a Class of Nonaffine Nonlinear Systems*”, IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, No. 1, January 2014
- [29] Min Wang, Cong Wang, Xiaoping Liu, “*Dynamic learning from adaptive neural control with predefined performance for a class of nonlinear systems*”, Information Sciences (2014)
- [30] Min Wang, Cong Wang, “*Learning From Adaptive Neural Dynamic Surface Control of Strict-Feedback Systems*”, IEEE Transactions on Neural Networks and Learning Systems 2014
- [31] Min Wang, Cong Wang, Liu Xiaoping, “*Learning from Adaptive Neural Control with Predefined Performance for a Class of Nonlinear Systems*”, Proceedings of the 33rd Chinese Control Conference July 28-30, Nanjing, China, 2014
- [32] A.J. Kurdila, Francis J. Narcowich, Joseph D. Ward “*Persistency of Excitation in Identification using Radial Basis Function Approximants*”, Siam J. Control and Optimization Vol.33 No. 2 pp. 625-642, March 1995
- [33] J.Park, J.W. Sandberg, “*Universal Approximation Using Radial-Basis-Function Networks*”, Neural Computation 3, 246-257 (1991)
- [34] R.M. Sanner, J.E. Slotine, “*Stable recursive identification using radial basis function networks*”, Proceedings of American Control Conference, Vol. 3, Chicago, pp. 1829-1833, 1992
- [35] Dimitry Gorinevsky, “*On the Persistency of Excitation in Radial Basis Function Network Identification of Nonlinear Systems*”, IEEE Transactions on Neural Networks, Vol.6, No.5, September 1995
- [36] Antonio Loria, Elena Panteley, “*Uniform exponential stability of linear time-varying systems: revisited*”, Systems & Control Letters, Volume 47, Issue 1, 16 September 2002
- [37] Charalampos P.Bechlioulis, George A. Rovithakis, “*Learning Structural Uncertainties of Nonlinear Systems with RBF Neural Networks via Persistently Exciting Control*”, Mediterranean Conference on Control and Automation, February 2, 2013.
- [38] C.P. Bechlioulis, G.A. Rovithakis, “*Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance*”, IEEE Transactions on Automatic Control, vol. 53, no.9, pp.2090-2099,2008

- [39] “Adaptive control with guaranteed transient and steady state tracking error bounds for strict feedback systems“, *Automatica*, vol. 45, no. 2, pp.532-538, 2009
- [40] “Prescribed performance adaptive control for multi-input multi-output affine in the control nonlinear systems“, *IEEE Transactions on Automatic Control*, vol. 55, no.5, pp. 1220-1226, 2010
- [41] C.P. Bechlioulis, G. Rovithakis, “Robust partial-state feedback prescribed performance control of cascade systems with unknown nonlinearities“, *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2224-2230, 2011
- [42] Paul Van Yee, “Regularized Radial Basis Function Networks: Theory and Applications to Probability Estimation, Classification and time series prediction“, Thesis, University of British Columbia (1989)
- [43] Bruce J. Torby, “Advanced Dynamics for Engineers“, p.269-270 (1984)
- [44] Louis N. Hand, Janet D. Finch, “Analytical Mechanics 1st Edition“, p.19 (2008)
- [45] Roger Penrose, “The Road to Reality: A Complete Guide to the Laws of the Universe“, (2007)
- [46] Mohammad Shahab, Ahmad Masoud, “2DOF Robotic Manipulator Control design & Simulation“, EE 656-Robotics & Control, 2008
- [47] J.Slotine and W.Li, “Applied Nonlinear Control“, Upper Saddle River, NJ: Prentice-Hall, 1991
- [48] Huyen T. Dinh, “Dynamic neural network-based robust control methods for uncertain nonlinear systems“, Dissertation, California, 2012.
- [49] Junmin and Cong Wang, “Fault detection for a class of Strict-feedback Systems Via Deterministic Learning“, July 15-17, 2012 – Dalian, China.