

Variable Structure Systems Theory in Computational Intelligence

Mehmet Önder Efe¹, Okyay Kaynak², and Xinghuo Yu³

¹ Carnegie Mellon University, Electrical and Computer Engineering Department
Pittsburgh, PA 15213-3890, U.S.A.

² Bogazici University, Electrical and Electronic Engineering Department Bebek,
80815, Istanbul, Turkey

³ Faculty of Informatics and Communication, Central Queensland University
Rockhampton QLD 4702, Australia

Abstract. Intelligence in the form of well-organized solutions to the ill-posed problems has been the primary focus of many engineering applications. The ever-increasing developments in data fusion, sensor technology and high-speed microprocessors made the design in digital domain with high performance. A natural consequence of the progression during the last few decades is the emergence of computationally intelligent systems. Neural networks and fuzzy inference systems constitute the core approaches of computational intelligence, whose methods have extensively been used in the applications extending from image/pattern recognition to identification and control of nonlinear systems. This chapter is devoted to the analysis and design of learning strategies in the context of variable structure systems. Several approaches are discussed in detail with special emphasis on the sliding mode control of nonlinear systems.

1 Introduction

Twentieth century has witnessed widespread innovations in all disciplines of engineering sciences. Two snapshots from early 1900s and late 1990s differ particularly in terms of the active role of humans in performing complicated tasks. The trend during the last century had the goal of implementing systems having some degrees of intelligence to cope with the problem specific difficulties that are likely to arise during the normal operation of the system. Today, it is apparent that the trend towards the development of autonomous machinery will maintain its importance as the tasks and the systems are getting more and more complicated. A natural consequence of the increase in the complexity of the task and physical hardware is to observe an ever-widening gap between the mathematical models and the physical reality to which the models correspond. Having this picture in front of us, what now becomes evident is the need for research towards the development of approaches having the capability of self-organization under the changing conditions of the task and the environment. Computational Intelligence (CI) is a framework offering various solutions to handle the complexity and the difficulties

of information-limited operating environments. The diversity in the solution space is a remarkable advantage that the designer utilizes either in the sense of algorithm-oriented manner or in the sense of architecture-oriented manner, hence, the result is an autonomous system exploiting these advantages.

Autonomy is one of the most important characteristics required from a computationally intelligent system. A basic requirement in this context is the ability to refresh and to refine the information content of the dynamics of the system. It therefore requires a careful consideration in the realm of engineering practice. From a systems and control engineering point of view, the designer is motivated by the time-varying nature of structural and environmental conditions to realize controllers that can accumulate the experience and improve the mapping precision [1-2]. Methodologies imitating the inference mechanism of the human brain are good in achieving the former and those imitating the massively interconnected structure of the human brain are good in achieving the latter. In the literature, the linguistic aspects of intelligence are discussed in the area Fuzzy Logic (FL) while the connectionist aspects are scrutinized in the area Neural Networks (NN). The integration of these methodologies that exploit the strength of each collectively and synergistically is a driving force to synthesize hybrid intelligent systems. Being not limited to what is mentioned, methods mimicking the process of evolution, which are discussed under the title Genetic Algorithms (GA), and those adapted from artificial intelligence constitute other branches of CI and fall beyond the focus of the approaches presented in this chapter.

NN are well known for their property of representing complex nonlinear mappings. Earlier works on the mapping properties of these architectures have shown that NN are universal approximators [3-5]. The mathematical power of intelligence is commonly attributed to the neural systems because of their structurally complex interconnections and fault tolerant nature. Various architectures of neural systems are studied in the literature. Feedforward and Recurrent Neural Networks (FNN, RNN) [6], Radial Basis Function Neural Networks (RBFNN) [1,6], dynamic neural networks [7], and Runge-Kutta neural networks [8] constitute typical topologically different models.

FL is the most popular constituent of the CI area since fuzzy systems are able to represent human expertise in the form of IF *antecedent* THEN *consequent* statements. In this domain, the system behavior is modeled through the use of linguistic descriptions. Although the earliest work by Prof. Lotfi Zadeh on fuzzy systems [9] has not been paid as much attention as it deserved in the early 1960s, since then the methodology has become a well-developed framework. The typical architectures of Fuzzy Inference Systems (FIS) are those introduced by Wang [10], Takagi and Sugeno [11] and Jang, Sun and Mizutani [1]. In [10], a fuzzy system having Gaussian membership functions, product inference rule and weighted average defuzzifier is constructed and has become the standard method in most applications. Takagi and Sugeno [11] change the defuzzification procedure where dynamic systems are used in

the defuzzification stage. The potential advantage of the method is that under certain constraints, the stability of the system can be studied. Jang et al [1] propose an adaptive neuro-fuzzy inference system, in which polynomials are used in the defuzzifier. This structure is commonly referred to as ANFIS in the related literature.

When the applications of NN and FL are considered the process of learning gains a vital importance. Although there is not a standard definition, the process of improving the future performance of the structures of CI by tuning the parameters can be described as learning. The approaches existing in the literature employ various techniques in achieving the desired parameter set (which is unknown), and require an iteratively evolving search mechanism. It should be noted that the most common technique that can be used in performing a suitable search operation in a multidimensional parameter space is based on the use of an appropriately defined cost function. Alternatively, the search procedure can be implemented without using the derivative information; such as is done by the use of methods adapted from the evolutionary computation, e.g. GAs, or random search methods [1].

Error Backpropagation (EBP) technique [12] and Levenberg-Marquardt (LM) optimization technique [13] are the frequently used techniques used for parameter adaptation in CI. Both approaches are based on the utilization of gradient information and necessitate the differentiability of the nonlinear activation functions existing in the architecture with respect to the parameter to be updated, and frequently utilize some heuristics for improved realization performance. These typically concern the selection of learning rate, momentum coefficient, and adaptive learning rate strategies in EBP or step-size considerations in LM technique. However, the problem of convergence or that of maintaining the bounded parameter evolution is an open problem associated with these approaches. More explicitly, the learning strategy is not protected against disturbances, which may excite the undesired internal modes of EBP or LM approaches. The multidimensionality of the problem is another difficulty in coming up with a thorough analysis distinguishing the useful training information and disturbance-related excitation signals. Since the ultimate goal of the design is to meet the performance specifications, reducing the adverse effects of the disturbances requires that the adopted learning dynamics should be robustified. This steers the designer to seek for methods known in the conventional design framework. From this point of view, a learning strategy based on Variable Structure Systems (VSS) theory constitutes a good candidate for eliminating the adverse effects of disturbances.

VSS with sliding modes were first proposed in early 1950s [14-15]. However, due to the implementation difficulties of high speed switching, it was not until 1970s that the approach received the attention it deserved. Sliding Mode Control (SMC) technique nowadays enjoys a wide variety of application areas; such as in general motion control applications and robotics, in

process control, in aerospace applications, and in power converters [16-19]. The main reason for this popularity is the attractive properties that SMC have, such as good control performance for nonlinear systems, applicability to Multi-Input-Multi-Output (MIMO) systems, and well established design criteria for discrete time systems. The most significant property of a sliding mode control system is its robustness. Loosely speaking, when a system is in a sliding mode, it is insensitive to parameter changes or external disturbances.

From a systems and control theoretic point of view, the primary characteristic of variable structure control is that the feedback signal is discontinuous, switching on one or more manifolds in the state space. When the state crosses each discontinuity surface, the structure of the feedback system is altered. Under certain circumstances, all motions in the neighborhood of the manifold are directed towards the manifold and thus a sliding motion on a predefined subspace of the state-space is established in which the system state repeatedly crosses the switching surface [20]. This mode has useful invariance properties in the face of uncertainties in the plant model and therefore is a good candidate for tracking control of uncertain nonlinear systems. The theory is well developed, especially for single-input systems in controller canonical form.

The theory of VSS with sliding modes has been studied intensively by many researchers. A recent comprehensive survey is given in [17] and various aspects of latest developments in VSS can be found in the chapters of this book. Motion control, especially in robotics, has been an area that has attracted particular attention and numerous reports have appeared in the literature [21-25]. One of the first experimental investigations that demonstrates the invariance property of a motion control system under a sliding mode is due to Kaynak et al [26].

In practical applications, a pure SMC approach suffers from the following disadvantages. Firstly, there is the problem of chattering, which is the high frequency oscillations of the controller output, brought about by the high speed (ideally at infinite frequency) switching necessary for the establishment of a sliding mode. In practical implementations, chattering is highly undesirable because it may excite unmodeled high frequency plant dynamics and this can result in unforeseen instabilities.

Secondly, a SMC based feedback loop is extremely vulnerable to measurement noise since the control input depends tightly on the sign of a measured quantity that is very close to zero [27]. Thirdly the SMC may employ unnecessarily large control signals to overcome the parametric uncertainties. Last but not least, there exists appreciable difficulty in the calculation of what is known as the equivalent control. A complete knowledge of the plant dynamics is required for this purpose [28]. To alleviate these difficulties, several modifications to the original sliding control law have been proposed [29], the most popular being the boundary layer approach, which is, in essence, the application of a high gain feedback when the motion of the system reaches

ϵ -vicinity of a sliding manifold [22,28]. This approach is based on the idea of the equivalence of the high gain systems and the systems with sliding modes [30]. Another variation of the scheme is called provident control that combines variable structure control and variable structure adaptation and performs hysteretic switching between the structures so as to avoid a sliding mode [31-32]. Both approaches are based on the calculation of the equivalent control, requiring a good mathematical model of the plant.

The essence of the discussion presented in this chapter is to integrate VSS technique and CI in an appropriate manner such that the difficulties of VSS approach are alleviated by intelligence and the mathematical intractability of intelligence is alleviated by VSS technique. Such a hybrid approach, particularly operating as the learning mechanism of CI architectures, is therefore a good candidate to represent the autonomous behavior of intelligent systems with a robustified learning performance.

2 A Functional Overview of Computationally Intelligent Architectures

2.1 Adaptive Linear Elements (ADALINEs)

Being categorized as the basic operation in all architectures of CI, ADALINE performs an inner product of two vectors, which is The output is a net sum in the case of NNs, or the response of the system in the cases of RBFNN, SFS, ANFIS. The vectors of interest are the adjustable parameter vector and the excitation input denoted by ϕ and \mathbf{u} respectively. The input-output relation can now be described as $\tau = \phi^T \mathbf{u}$, where τ is the scalar output. Clearly, the applications requiring multiple outputs τ would be a vector while ϕ would be a matrix of appropriate dimensions.

2.2 Feedforward Neural Networks (FNNs)

FNNs constitute a class of NN structures in which the data flow is from input to the output and no feedback connections are allowed. Because of the structural diversity of neural models, this discussion is devoted to the architecture and the mathematical representation of FNN structure, which is discussed from the point of control engineering. The architecture of a typical FNN is illustrated in Figure 1, in which the neural network has three layers implying the sufficiency for realizing any continuous mapping to a desired degree of accuracy as long as the hidden layer contains sufficiently many neurons [3-5]. The number of neurons in the hidden layer is a design variable and is mostly determined either by trial and error or by empirical results.

Functionally, $\mathbf{o} = \psi_h(W_h \mathbf{u})$ and $\boldsymbol{\tau} = \psi_0(W_0 \mathbf{o})$, where ψ_h and ψ_0 stand for the vectors of nonlinear activation functions for the hidden layer and the output layer respectively. Adaptation is carried out on the adjustable weights

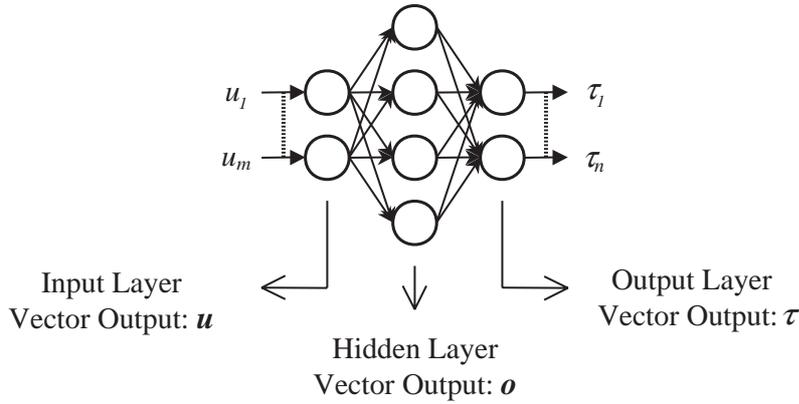


Fig. 1. Structure of a FNN

contained in W_h and W_0 matrices. In most applications of NNs, hyperbolic tangent or sigmoid functions are used in ψ_h whereas the selection of ψ_0 is generally a linear function of its argument, e.g. $\psi_0(x) = x$. The standard approach for tuning the parameters of FNNs is EBP or LM techniques [12-13].

Information contained in such a nonlinear map is distributed over its architectural constituents, i.e. neurons, such that a local failure in the structure can be tolerated because of the parametric redundancy existing in the structure, which is an analogue of the fault tolerance in biological systems. More explicitly, the task can be redistributed upon death of neurons forming a local infrastructure of a massive network.

2.3 Radial Basis Function Neural Networks (RBFNNs)

In the literature, RBFNNs are generally considered as a smooth transition between FL and NNs. Structurally, a RBFNN is composed of receptive units (neurons) which act as the operators providing the information about the class to which the input signal belongs. If the aggregation method, number of receptive units in the hidden layer and the constant terms are equal to those of a FIS, then there exists a functional equivalence between RBFNN and FIS [1]. Although the architectural view of a RBFNN is very similar to that of a FNN illustrated in Figure 1, the hidden neurons of a RBFNN possess basis functions to characterize the partitions of the input space. Each neuron in the hidden layer provides a degree of membership value for the input pattern with respect to the basis vector of the receptive unit itself. The output layer is comprised of linear neurons. NN interpretation makes RBFNN useful in incorporating the mathematical tractability, especially in the sense of propagating the error back through the network, while the FIS interpretation enables the incorporation of the expert knowledge into the

training procedure. The latter is of particular importance in assigning the initial value of the network's adjustable parameter vector to a vector that is to be sought iteratively. Expectedly, this results in faster convergence in parameter space.

Mathematically, $o_i = \prod_{j=1}^m \psi_{ij}(u_j)$ and a common choice for the hidden layer activation function is the Gaussian curve described as $\psi_{ij}(\mathbf{u}) = \exp\{-(u_j - c_{ij})^2/\sigma_{ij}^2\}$, where c_{ij} and σ_{ij} stand for the center and the variance of the i^{th} neuron's activation function qualifying the j^{th} input variable. The output of the network is evaluated through the inner product of the adjustable weight vector denoted by Φ and the vector of hidden layer outputs, i.e. $\tau = \Phi^T \mathbf{o}$, which is just as in the case of output evaluation in ADALINEs. Clearly the adjustable parameter set of the structure is composed of $\{\mathbf{c}, \sigma, \Phi\}$ triplet.

2.4 Standard Fuzzy Systems (SFSs)

Contrary to what is postulated in the realm of predicate logic, representation of knowledge by fuzzy quantities can provide extensive degrees of freedom if the task to be achieved can better be expressed in words than in numbers. The concept of fuzzy logic in this sense can be viewed as a generalization of binary logic and refers to the manipulation of knowledge with sets, whose boundaries are unsharp [33]. Therefore the paradigm offers a possibility of designing intelligent controllers operating in an environment, in which the conditions are inextricably intertwined, subject to uncertainties and impreciseness.

Understanding the information content of fuzzy logic systems is based on the subjective judgements, intuitions and the experience of an expert. From this point of view, a suitable way of expressing the expert knowledge is the use of *IF antecedent THEN consequent* rules, which can easily evaluate the necessary action to be executed for the current state of the system under investigation.

Structurally, a fuzzy controller is comprised of five building blocks, namely, fuzzification, inference engine, knowledge base, rule base, and defuzzification. Since the philosophy of the fuzzy models is based on the representation of knowledge in fuzzy domain, the variables of interest are graded first. This grading is performed through the evaluation of membership values of each input variable in terms of several class definitions. According to the definition of a membership function, how the degree of confidence changes over the domain of interest is characterized. This grading procedure is called fuzzification. In the knowledge base, the parameters of membership functions are stored. Rule base contains the cases likely to happen, and the corresponding actions for those cases through linguistic descriptions, i.e. the IF-THEN statements. The inference engine emulates the expert's decision making in interpreting and applying knowledge about how the best fulfillment of the task is achieved. Finally, the defuzzifier converts the fuzzy decisions back onto the crisp domain [34].

SFS architecture that has been proposed by Wang [35] uses algebraic product operator for the aggregation of the rule premises and bell-shaped membership functions denoted by μ . The overall representation of SFS structure is given in (1), in which R and m stand for the number of rules contained in the rule base and the number of inputs of the structure.

$$\tau = \sum_{i=1}^R f_i \left(\frac{\prod_{j=1}^m \mu_{ij}(u_j)}{\sum_{i=1}^R \prod_{j=1}^m \mu_{ij}(u_j)} \right) \quad (1)$$

with i^{th} rule as: IF u_1 is U_{1i} AND u_2 is U_{2i} AND ... AND u_m is U_{mi} THEN $f_i = \phi_i$. In the IF part of this representation, the lowercase variables denote the inputs and the uppercase variables stand for the fuzzy sets corresponding to the domain of each linguistic label. The THEN part is comprised of the prescribed decision in the form of a scalar number denoted by ϕ_i . Clearly, the adjustable parameters of the structure are comprised of the parameters of the membership functions together with the defuzzifier parameters ϕ_i . Another common feature of the representation in (1) is the linearity of the output in the defuzzifier parameters.

2.5 Adaptive Neuro-Fuzzy Inference Systems (ANFIS)

Adaptive neuro-fuzzy inference systems are synthesized by an appropriately integrating the neural and fuzzy system interpretations. The resulting hybrid combination therefore inherits the numeric power of NN as well as the verbal power of FL [1,36]. An ANFIS structure having m -inputs and single output with product inference rule and first order Sugeno model can be described as in (1) with f_i being described as in the rule consequent. The structural view of such a system is illustrated in Figure 2, in which \tilde{N} stands for the normalization operator seen as the last term of (1).

The rule structure for an ANFIS utilizing first order Sugeno model has the following representation: IF u_1 is U_{1i} AND u_2 is U_{2i} AND ... AND u_m is U_{mi} THEN $f_i = \phi_{i,1}u_1 + \dots + \phi_{i,m}u_m + \phi_{i,m+1}$. When the consequent part of the rule structure is compared with that of rules in SFS architecture, it is seen that the polynomial representation of the decision introduces higher parametric flexibility extending the realization capability. Being not confined to what is discussed above, depending on the requirements of the problem in hand, the designer can choose higher order polynomials to improve the realization accuracy. When the issue of parameter tuning in ANFIS is considered the well-known gradient approaches as well as the method of least mean squares or VSS based approaches can easily be utilized.

ANFIS structure has been utilized with gradient based training strategies for identification of nonlinear systems [37] and with VSS based training strategies for variable structure control of motion control systems. In [1], an in-depth discussion is given with numerous examples on the use of ANFIS structure.

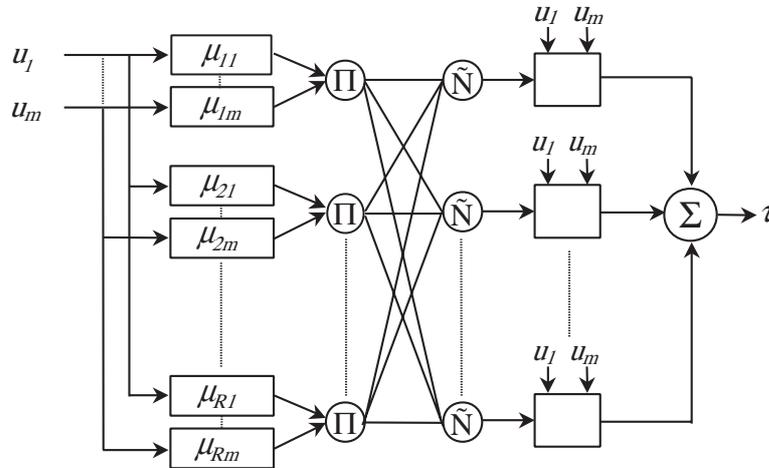


Fig. 2. Structure of an ANFIS

3 VSS Based Parameter Tuning in Intelligent Control Systems

The studies reporting the use of VSS for parameter tuning in CI by Sanner and Slotine [38], and Sira-Ramirez and Colina-Morles [39] have been the stimulants, which proved that the robustness feature of VSS could be exploited in the training of the architectures of CI. These studies pioneered a vast majority of researchers working on VSS and CI. Sanner and Slotine considered the training of GRBFNN which has certain degrees of analytical tractability in explaining the stability issues, and Sira-Ramirez et al have shown the use of ADALINEs with a VSS based learning strategy. As an illustrative example, the inverse dynamics identification of a Kapitsa pendulum has been demonstrated together with a thorough analysis towards the handling of disturbances. Hsu and Real [40-41] demonstrate the use of VSS with Gaussian NNs, Yu et al [42] introduces the dynamic uncertainty adaptation of what is proposed in [39], and demonstrate the performance of the scheme on the Kapitsa pendulum. Parma et al [43] use the VSS technique in parameter tuning process of multilayer perceptron. Latest studies towards the integration of VSS and CI have shown that the tuning can be implemented in dynamic weight filter neurons [44], in parameters of a controller [45]. A different viewpoint towards this integration is due to Efe et al [46-47], which has the goal of reducing the adverse effects of noise driven parameter tuning activity in gradient techniques. The key idea in these works is to mix the two training signals in a weighted average sense. A good deal of review is provided in the recent survey of Kaynak et al [48]. The survey illustrates how

VSS can be used for training in CI as well as how CI can be utilized for the tuning of parameters in conventional VSS.

In what follows, the use of VSS approach for intelligent control of nonlinear systems is presented together with the analytical details wherever required. The emphasis is mainly on the works presented in [44-45] with the authors' latest research outcomes towards the direction of control engineering.

3.1 Control System Structure

Consider the feedback loop illustrated in Figure 3, in which a subscript d denotes the desired value of the relevant quantity. Furthermore, it is shown in the figure that if a supervisor provides the desired controller outputs, one might evaluate the error on the control signal denoted by s_c .

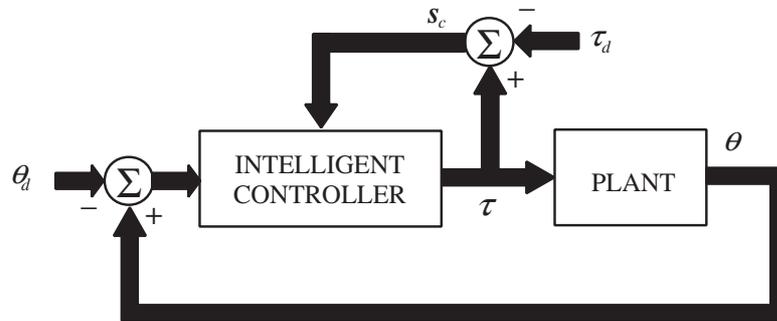


Fig. 3. Block diagram of the control system

The plant shown in Figure 3 is assumed to have the structure described in (2), in which θ and τ are $(r_1 + r_2 + \dots + r_n) \times 1$ -dimensional state vector and $n \times 1$ -dimensional input vector. The system of (2) with these vectors can be restated as $\dot{\theta} = f_p(\theta) + D\tau$.

$$\theta_i^{(r_i)} = f_{p_i}(\theta) + \sum_{j=1}^n d_{ij}\tau_j \quad i = 1, 2, \dots, n \quad (2)$$

The design problem is to enforce the behavior of the system towards the desired response, which is known but the control signal (τ_d) resulting in which is unavailable. Therefore, the solution to this problem is a search towards the synthesis of such a signal iteratively by an intelligent controller. Assuming that the intelligent controller in Figure 3 is composed of n individual controllers, the i^{th} one of which is to construct the i^{th} component of input vector τ , the j^{th} entry of the error vector driving this sub-controller can be

given as $e_i^{(j)} = \theta_i^{(j)} - \theta_{di}^{(j)}$. Apparently, this component is the j^{th} derivative of the relevant state component.

3.2 Conventional VSS Design - An Overview

Consider the vector of sliding surfaces for the system in (2): $\mathbf{s}_p(\mathbf{e}) = G\mathbf{e} = G(\boldsymbol{\theta} - \boldsymbol{\theta}_d)$. The widespread selection of the matrix G is such that the i^{th} sliding surface function has the form

$$s_{pi}(e_i) = \left(\frac{d}{dt} + \lambda_i\right)^{r_i-1} e_i \tag{3}$$

in which, λ_i is a strictly positive constant. Let V_p be a candidate Lyapunov function given as

$$V_p(\mathbf{s}_p) = \frac{1}{2} \mathbf{s}_p^T \mathbf{s}_p \tag{4}$$

If the prescribed control signal satisfies $\dot{V}_p(\mathbf{s}_p) = -\mathbf{s}_p^T \Xi \text{sgn}(\mathbf{s}_p)$, the negative definiteness of the time derivative of the Lyapunov function in (4) is ensured. In above, Ξ is a positive definite diagonal matrix of dimension $n \times n$. More explicitly, $\mathbf{s}_p^T \dot{\mathbf{s}}_p = -\mathbf{s}_p^T \Xi \text{sgn}(\mathbf{s}_p)$ must hold true to drive the error vector towards the sliding hypersurface. On the other hand, the use of $\dot{\mathbf{s}}_p = -G\dot{\boldsymbol{\theta}}_d + G(\mathbf{f}_p(\boldsymbol{\theta}) + D\boldsymbol{\tau})$ leads to the following control signal:

$$\boldsymbol{\tau} = -(GD)^{-1} (G\mathbf{f}_p(\boldsymbol{\theta}) - G\dot{\boldsymbol{\theta}}_d) - (GD)^{-1} \Xi \text{sgn}(\mathbf{s}_p) \tag{5}$$

in which, the first term is the equivalent control term and the second term is the corrective control term. For the existence of the mentioned components, the matrix GD must not be rank deficient. In the literature, equivalent control is considered as the low frequency (average) component of the control signal. Because of the discontinuity on the sliding surface, the corrective term brings a high rate component [20,25]. If $\mathbf{e}(0) = \mathbf{0}$, the tracking problem can be considered as keeping \mathbf{e} on the sliding surface, however, for nonzero initial conditions, the strategy must enforce the state trajectories towards the sliding surface, which is ensured by the negative definiteness of the time derivative of the Lyapunov function as in (4). For the case of nonzero initial conditions, the phase until the error vector hits the sliding surface is called the reaching mode, the dynamic characteristics of the system during which is determined by the control strategy adopted. Application of the control input formulated in (5) imposes the dynamics described as $\dot{\mathbf{s}}_p = -\Xi \text{sgn}(\mathbf{s}_p)$, which clearly enforces the error vector towards the sliding surface. Once the sliding surface is reached, the value of (3) becomes zero; and this enforces the error vector to move towards the origin.

Aside from the practical difficulties of conventional VSS schemes, the control signal in (5) is applicable if a nominal representation of the system under control is available. In the next subsection, a method for obtaining the error on the control signal is presented for unknown systems of structure (2).

3.3 Calculation of the Control Error

Remark 3.1: The VSS task is achievable if the dynamics of the system in (2) is totally known or if the nominal system is known with the bounds of the uncertainties. It must be noted that to satisfy the matching conditions, the disturbances and uncertainties are always assumed to enter the system through the control channels [17]. When the conventional VSS strategy is applied to the system of (2), we call the resulting behavior as the *target VSS* and the input vector leading to it as the *target control sequence* ($\boldsymbol{\tau}$), which is described in (5). If the functional form of the vector function \boldsymbol{f}_p is not known, it should be obvious that the target control sequence cannot be constructed by following the traditional VSS design approaches.

Definition 3.2: Given an uncertain plant, which has the structure described as in (2), and a command trajectory vector $\boldsymbol{\theta}_d(t)$ for $t \geq 0$, the input sequence satisfying the following vector differential equation is defined to be the *idealized control sequence* denoted by $\boldsymbol{\tau}_d$, and the vector differential equation itself is defined to be the *reference SMC model*.

$$\dot{\boldsymbol{\theta}}_d = \boldsymbol{f}_p(\boldsymbol{\theta}_d) + D\boldsymbol{\tau}_d \quad (6)$$

Mathematically, the existence of such a model and the sequence means that the system of (2) perfectly follows the command trajectory vector if both the idealized control sequence is known and the initial conditions are set as $\boldsymbol{\theta}(t=0) = \boldsymbol{\theta}_d(t=0)$, more explicitly $\boldsymbol{e}(t) \equiv 0$ for $t \geq 0$. Undoubtedly, such an idealized control sequence will not be a norm-bounded signal when there are step-like changes in the vector of command trajectories or when the initial errors are nonzero. It is therefore that the reference SMC model is an abstraction due to the limitations of the physical reality, but the concept of idealized control sequence should be viewed as the synthesis of the command signal $\boldsymbol{\theta}_d$ from the time solution of the differential equation set in (6).

Fact 3.3: Based on the Lyapunov stability results of the previous subsection, if the target control sequence formulated in (5) were applied to the system of (2), the idealized control sequence would be the steady state solution of the control signal, i.e. $\lim_{t \rightarrow \infty} \boldsymbol{\tau} = \boldsymbol{\tau}_d$. However, under the assumption of the achievability of the VSS task, the difficulty here is again the unavailability of the functional form of the vector function \boldsymbol{f}_p . Therefore, the aim in this subsection is to discover an equivalent form of the discrepancy between the control applied to the system and its target value by utilizing the idealized control viewpoint. This discrepancy measure is denoted by $\boldsymbol{s}_c = \boldsymbol{\tau} - \boldsymbol{\tau}_d$ and is of $n \times 1$ dimension.

If the target control sequence of (5) is rewritten by using (6), one gets

$$\boldsymbol{\tau} = -(GD)^{-1} \left(G\boldsymbol{f}_p(\boldsymbol{\theta}) - G(\boldsymbol{f}_p(\boldsymbol{\theta}_d) + D\boldsymbol{\tau}_d) + \Xi \operatorname{sgn}(\boldsymbol{s}_p) \right)$$

$$\begin{aligned}
 &= -(GD)^{-1} \left(G\mathbf{f}_p(\boldsymbol{\theta}) - G\mathbf{f}_p(\boldsymbol{\theta}_d) + \Xi \operatorname{sgn}(\mathbf{s}_p) \right) + \boldsymbol{\tau}_d \\
 &= -(GD)^{-1} \left(G\boldsymbol{\Delta}\mathbf{f}_p(\boldsymbol{\theta}) + \Xi \operatorname{sgn}(\mathbf{s}_p) \right) + \boldsymbol{\tau}_d
 \end{aligned} \tag{7}$$

The target control sequence becomes identical to the idealized control sequence, i.e. $\boldsymbol{\tau} \equiv \boldsymbol{\tau}_d$, as long as $G\boldsymbol{\Delta}\mathbf{f}_p(\boldsymbol{\theta}) + \Xi \operatorname{sgn}(\mathbf{s}_p) = \mathbf{0}$ holds true. However, this condition is of no practical importance as we do not have the analytic form of the vector function \mathbf{f}_p . Therefore, one should consider this equality as an equality to be enforced instead of an equality that holds true all the time, because its implication is $\mathbf{s}_c = \mathbf{0}$ and is the aim of the design. It is obvious that to enforce this to hold true will let us synthesize the target control sequence, which will ultimately converge to the idealized control sequence by the adaptation algorithm yet to be discussed. Consider the time derivative of the vector of sliding surfaces

$$\begin{aligned}
 \dot{\mathbf{s}}_p(\mathbf{e}) &= G\dot{\mathbf{e}} \\
 &= G(\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}_d) \\
 &= G\left(\mathbf{f}_p(\boldsymbol{\theta}) + D\boldsymbol{\tau} - \mathbf{f}_p(\boldsymbol{\theta})_d - D\boldsymbol{\tau}_d\right) \\
 &= G\left(\boldsymbol{\Delta}\mathbf{f}_p + D(\boldsymbol{\tau} - \boldsymbol{\tau}_d)\right) \\
 &= G\left(\boldsymbol{\Delta}\mathbf{f}_p + D\mathbf{s}_c\right)
 \end{aligned} \tag{8}$$

Utilizing $G\boldsymbol{\Delta}\mathbf{f}_p + \Xi \operatorname{sgn}(\mathbf{s}_p) = \mathbf{0}$ in (8) and solving for \mathbf{s}_c yields the following relation:

$$\mathbf{s}_c = (GD)^{-1} \left(\dot{\mathbf{s}}_p + \Xi \operatorname{sgn}(\mathbf{s}_p) \right) = \boldsymbol{\tau} - \boldsymbol{\tau}_d \tag{9}$$

Remark 3.4: The reader must here notice that the application of $\boldsymbol{\tau}_d$ to the system of (2) with zero initial errors will lead to $\mathbf{e}(t) \equiv \mathbf{0}$ for $\forall t \geq 0$, on the other hand, the application of $\boldsymbol{\tau}$ to the system of (2) will lead to $\mathbf{s}_p = \mathbf{0}$ for $\forall t \geq t_h$, where t_h is the hitting time, and the origin will be reached according to the dynamics of the sliding surface. Therefore, the adoption of (9) as the equivalent measure of the control error loosens $\mathbf{e}(t) \equiv \mathbf{0}$ for $\forall t \geq 0$ requirement and introduces all trajectories in the error space to tend to the sliding hypersurface, i.e. $G\boldsymbol{\Delta}\mathbf{f}_p + \Xi \operatorname{sgn}(\mathbf{s}_p) = \mathbf{0}$ is enforced. Consequently, the tendency of the control scheme will be to generate the target VSS sequence of (5) without requiring the analytical details of the plant.

Now consider the ordinary feedback control loop illustrated in Figure 3, and define the following Lyapunov function, which is a measure of how well the controller performs:

$$V_c(\mathbf{s}_c) = \frac{1}{2} \mathbf{s}_c^T \mathbf{s}_c \tag{10}$$

Remark 3.5: An adaptation algorithm ensuring $\dot{V}_c(\mathbf{s}_c) < 0$ when $\mathbf{s}_c \neq \mathbf{0}$ enforces $G\Delta\mathbf{f}_p + \Xi \operatorname{sgn}(\mathbf{s}_p) = \mathbf{0}$ and creates the predefined sliding regime after a reaching mode lasting until the hitting time denoted by t_h , beyond which $\mathbf{s}_c = \mathbf{0}$ as the system is in the sliding regime. If $\dot{V}_c(\mathbf{s}_c) < 0$ when $\mathbf{s}_c \neq \mathbf{0}$, then $\lim_{t \rightarrow t_h} V_c = 0 \iff \lim_{t \rightarrow t_h} \|\mathbf{s}_c\| = 0 \iff \lim_{t \rightarrow t_h} \|\dot{\mathbf{s}}_p + \Xi \operatorname{sgn}(\mathbf{s}_p)\| = 0$. Note that the meaning of $\mathbf{s}_c = \mathbf{0}$ is now equivalent to $\mathbf{s}_p = \mathbf{0}$ by Remark 3.4, therefore the limits above are evaluated as $t \rightarrow t_h$.

3.4 Parameter Tuning based on a Single-Term Lyapunov Function

If the architectures introduced in the second section are utilized for the purpose of control, without loss of generality, the output of the i^{th} controller can be restated as $\tau_i = \phi_i^T \Omega_i$, where Ω_i is the vector of signals exciting the adjustable parameters denoted by ϕ_i . Therefore the algorithm discussed here is applicable to ADALINE, GRBFNN, SFS and ANFIS architectures. Furthermore, the Lyapunov function in (10) constitutes the basis of the design.

In order not to be in conflict with the physical reality, the designer must impose $\|\phi_i\| \leq B_{\phi_i}$, $\|\Omega_i\| \leq B_{\Omega_i}$, $\|\dot{\Omega}_i\| \leq B_{\dot{\Omega}_i}$, and $\|\dot{\tau}_{id}\| \leq B_{\dot{\tau}_{id}}$ the truth of which state that the adjustable parameters of the controller, the time derivative of the signal exciting the adjustable parameter set and the time derivative of the idealized output of the controller remain bounded. Note that in Definition 3.2, we stated that there may not be a finite $B_{\dot{\tau}_{id}} \in \mathfrak{R}$ even in some realistic situations like nonzero initial errors, however, the practical meaning of imposing $\|\dot{\tau}_{id}\| \leq B_{\dot{\tau}_{id}}$ will lead us to the construction of an approximation of the idealized control sequence and the requirement of $e(t) \equiv \mathbf{0}$ for $\forall t \geq 0$ must therefore be loosened.

Theorem 3.6: For the i^{th} subsystem of the system described in (2), adopting the controller of structure $\tau_i = \phi_i^T \Omega_i$, the adaptation of the controller parameters as described in (11) enforces the value of the i^{th} component of control discrepancy vector (s_{c_i}) to zero.

$$\dot{\phi}_i = -\frac{\Omega_i}{\Omega_i^T \Omega_i} k_i \operatorname{sgn}(s_{c_i}) \quad (11)$$

where, k_i is a sufficiently large positive constant satisfying $k_i > B_{\phi_i} B_{\dot{\Omega}_i} + B_{\dot{\tau}_{id}}$. The adaptation mechanism in (11) drives an arbitrary initial value of s_{c_i} to zero in finite time denoted by t_{hi} satisfying the inequality in (12).

$$t_{hi} \leq \frac{|s_{c_i}(0)|}{k_i - (B_{\phi_i} B_{\dot{\Omega}_i} + B_{\dot{\tau}_{id}})} \quad (12)$$

Proof: See Sira-Ramirez et al [39] and Efe et al [45].

An important feature of this approach is the fact that the controller parameters evolve bounded as assumed initially. The details of the bounded parametric evolution analysis can be found in [42,45].

3.5 Parameter Tuning based on a Two-Term Lyapunov Function

Similar to what is initially designated in the previous subsection, the output of the i^{th} controller is described as $\tau_i = \phi_i^T \Omega_i$. In addition to the stated boundedness conditions the truth of $\|\Omega_i\| \leq B_{\Omega_i}$ is imposed. Consider the Lyapunov function given in (13), in which μ and ρ are the weights to be selected by the designer.

$$V = \mu V_{c_i} + \rho \frac{1}{2} \left\| \frac{\partial V_{c_i}}{\partial \phi_i} \right\|^2 \quad \text{with} \quad V_{c_i} = \frac{1}{2} s_{c_i}^2 \quad (13)$$

Theorem 3.7: If the adaptation strategy for the adjustable parameters of the i^{th} controller is chosen as

$$\dot{\phi}_i = -k_i \left(\mu I + \rho \frac{\partial^2 V_{c_i}}{\partial \phi_i \partial \phi_i^T} \right)^{-1} \text{sgn} \left(\frac{\partial V_{c_i}}{\partial \phi_i} \right) \quad (14)$$

with k_i is a sufficiently large constant satisfying $k_i > (\mu B_{\phi_i} + \rho B_{\Omega_i}) B_{\Omega_i}$, then the negative definiteness of the time derivative of the Lyapunov function in (13) is ensured.

Proof: See Efe [49].

3.6 A Generalization of EBP and LM Techniques in the Context of VSS

A recent contribution towards the generalization of EBP and LM techniques is due to Yu et al [50]. The approach postulated is applicable to all architectures discussed in the second section and is based on the Lyapunov function given in (15).

$$V(J_i, \phi_i) = \mu J_i + \rho \frac{1}{2} \left\| \frac{\partial J_i}{\partial \phi_i} \right\|^2 \quad (15)$$

where $J_i = \gamma^{-1} \int_{t-\gamma}^t s_{c_i}(\sigma) d\sigma$ with γ being the length of a time window to evaluate the training efficiency [51-52].

Theorem 3.8: For a computationally intelligent structure whose input-output relationship is $\tau_i(t) = \mathfrak{F}(\phi_i(t), \mathbf{u}_i(t))$, if

- (a) $\frac{\partial J_i}{\partial t} < 0$ and
- (b) The parameter adaptation rule is

$$\dot{\phi}_i = \begin{cases} -\left(\mu I + \rho \frac{\partial^2 J_i}{\partial \phi_i \partial \phi_i^T}\right)^{-1} \left(\frac{\frac{\partial J_i}{\partial \phi_i^T}}{\left\|\frac{\partial J_i}{\partial \phi_i}\right\|^2}\right)^* & \\ \left(\mu \frac{\partial J_i}{\partial t} + \rho \frac{\partial J_i}{\partial \phi_i} \frac{\partial^2 J_i}{\partial t \partial \phi_i^T} + \zeta \left\|\frac{\partial J_i}{\partial \phi_i}\right\|^2 + \eta J_i\right) & \text{if } \left\|\frac{\partial J_i}{\partial \phi_i}\right\| \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Then $\frac{\partial J_i}{\partial \phi_i}$ tends to zero asymptotically.

Proof: See Yu et al [50].

The formulation of J_i is particularly useful for on line training and continuous time learning. However, for discrete data, since the evaluation of errors can only be done at discrete instants of time, J_i at time t_k can be defined as $J_i(t = t_k) = \lim_{\gamma \rightarrow 0} \gamma^{-1} \int_{t-\gamma}^t s_{c_i}(\sigma) d\sigma = s_{c_i}(t_k)$. The conventional gradient descent learning algorithm can be now obtained by setting $\rho = 0$ and $\eta = 0$. Since $\frac{\partial^2 J_i(t=t_k)}{\partial t \partial \phi_i^T} = 0$ one obtains the law in (17), whose learning rate in the conventional sense is $\eta^{-1}\zeta$.

$$\dot{\phi} = -\eta^{-1}\zeta \frac{\partial J_i(t = t_k)}{\partial \phi^T} \quad (17)$$

The Gauss-Newton algorithm can be obtained by setting $\mu = 0$ and $\eta = 0$. Since $\frac{\partial J_i(t=t_k)}{\partial t} = 0$ and $\frac{\partial^2 J_i(t=t_k)}{\partial t \partial \phi_i^T} = 0$, from (16) one gets the law in (18).

$$\begin{aligned} \dot{\phi}_i &= \left(\sigma \frac{\partial^2 J_i(t = t_k)}{\partial \phi_i \partial \phi_i^T}\right)^{-1} \left(\zeta \frac{\partial J_i(t = t_k)}{\partial \phi_i^T}\right) \\ &= -\sigma^{-1}\zeta \left(\frac{\partial^2 J_i(t = t_k)}{\partial \phi_i \partial \phi_i^T}\right)^{-1} \left(\frac{\partial J_i(t = t_k)}{\partial \phi_i^T}\right) \end{aligned} \quad (18)$$

Similarly, the LM algorithm can be easily obtained by setting $\eta = 0$. Since $\frac{\partial J_i(t=t_k)}{\partial t} = 0$ and $\frac{\partial^2 J_i(t=t_k)}{\partial t \partial \phi_i^T} = 0$, from (16) the law in (19) is obtained.

$$\dot{\phi}_i = -\left(\mu I + \sigma \frac{\partial^2 J_i(t = t_k)}{\partial \phi_i \partial \phi_i^T}\right)^{-1} \left(\zeta \frac{\partial J_i(t = t_k)}{\partial \phi_i^T}\right) \quad (19)$$

3.7 Practical Issues

The analysis and the design approach presented so far have tried to illuminate the VSS based training problem from a theoretical perspective. In this subsection, we discuss several issues related to the practical applications of the discussed methodologies.

Chattering Since the control decision during the sliding mode is tightly dependent to the sign of a measured quantity being noisy and very close to zero, the decision along the sliding manifold exhibits sensitivity to noise on the observations. Among many alternatives available [17,28,53], a common approach to eliminate the chattering is to smooth the sign function, which corresponds to introduce a boundary layer [28]. A widespread choice is the following approximation for the $\text{sgn}(\cdot)$ function.

$$\text{sgn}(x) \cong \frac{x}{|x| + \delta} \quad (20)$$

where δ determines the sharpness around the origin. Since the function in (20) is not discontinuous at the origin, the decision mechanism softly switches inside the boundary layer.

Actuation Speed Another important issue is the actuation speed of the system under control, i.e. the ability to respond to what is imposed timely. Since the details concerning the dynamic model of the plant under control are assumed to be unavailable, what causes a difficulty from a practical point of view is the selection of the matrix Ξ , which characterizes the behavior during the reaching mode. The values of this quantity can only be set by trial-and-error due to the lack of system-specific details.

Obtaining the Equivalent Error from the Observed Data Lastly in this subsection, we focus on the construction of the \mathbf{s}_c of (9), which requires the differentiation of \mathbf{s}_p . A suitable approach is to filter the measured values of \mathbf{s}_p and differentiate afterwards. Denote S as the Laplace variable, and use the linear dynamic system given as

$$H(S) = \frac{\alpha S}{Q(S)} \quad (21)$$

where $Q(0) = \alpha > 0$ and $\text{Real}\{\text{roots}(Q(S))\} < 0$. The order of the denominator polynomial and the locations of the roots are left to the designer, because these issues require several trials to refine the selections and are subject to the application together with its operating environment. It should be noted that the cost of the information loss by using such a filter, whose input is \mathbf{s}_p and output is an estimate of $\dot{\mathbf{s}}_p$, is a matter of how robust the devised control algorithm is. More explicitly, the separation of the noise and the actual value of \mathbf{s}_p leads to a corruption on \mathbf{s}_p , and when differentiated afterwards, some valuable information is lost together with the elimination of the noise component. Here it is assumed that the mentioned loss causes an uncertainty, which enters the system through the control channels, and which is particularly effective during the sliding mode; and this uncertainty can be alleviated if it falls within the limits allowing the maintenance of the invariance during the sliding mode [17].

Computational Burden One of the factors qualifying the physical implementability of control schemes is the number of computations to be performed by the controller. In this part, a discussion of the computational burden of the tuning mechanisms is presented. It should be noted that the structure of the controller adopted strictly influences the required number of *floating point operations* (flops) between the two consecutive sampling instants. Therefore, the discussion given here focuses on the ADALINE controller, as it constitutes a basis of all structures. If an ANFIS structure is to be used, the designer must consider the extra calculations to generate the vector signal exciting the adjustable parameter set of the defuzzifier.

Another point to clarify is the computational complexity due to the approach postulated in Theorem 3.8, whose practical applications generally subject to the following: the cost function J_i is evaluated at the discrete instants of time and it does not depend explicitly on time, i.e. $\frac{\partial J_i(t=t_k)}{\partial t} = 0$ and $\frac{\partial^2 J_i(t=t_k)}{\partial t \partial \phi_i^T} = 0$.

Figure 4 illustrates a bar graph composed of triplets. The leftmost component represents the flops required to evaluate the ADALINE output and to adjust its parameters once by utilizing the method discussed in the subsection 3.4. The middle and the rightmost components stand for the required number of flops for the methods presented in subsections 3.5 and 3.6 respectively. It is clear from the figure that the complexity due to the first approach is considerably smaller than the other two as the order of the subsystem under control increases. This fact is primarily because of the matrix inversion to be performed at each step. However, the set of criteria qualifying the performance of an intelligent control system is strictly dependent upon the application specific details, which does not give a clue in choosing a tuning mechanism. Therefore, the designer is encouraged to try the alternatives in discovering the one performing the best.

3.8 Summary

What we have discussed so far have illuminated the design considerations at microscopic levels forming the whole picture. When implementing the control system with a VSS based tuning mechanism updating the parameters of an intelligent controller, one has to remember that the plant is in an ordinary feedback loop as illustrated in Figure 3. Having decided on the controller structure, the error vector is processed until the control to be applied is obtained. Since the desired control inputs are unavailable, using the error measure given in (9), the similarity between the applied control and the target control sequence is qualified, then the parameter tuning is performed according to the chosen tuning strategy.

A particular difference in applying the ADALINE structure as the controller with (11) and (14) is that the controller input vector is formed by augmenting the error vector, which is of dimension $r_i \times 1$, with a constant

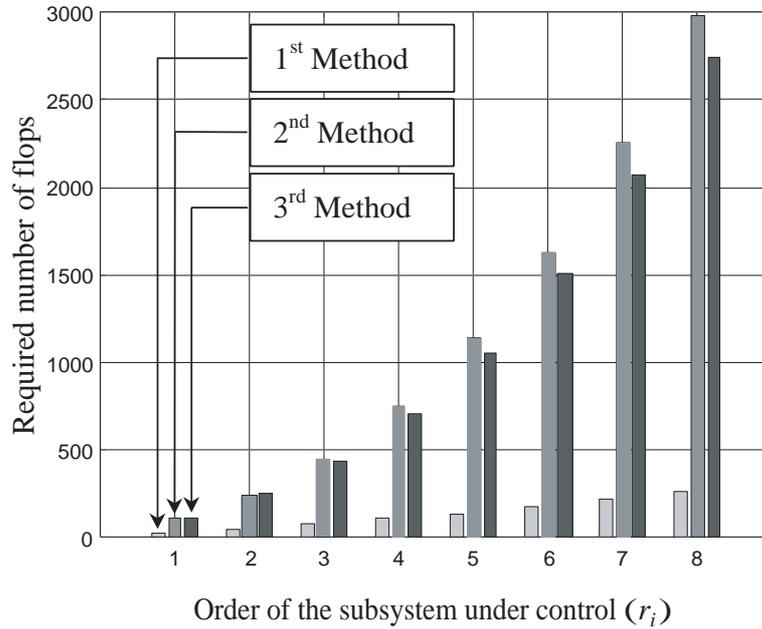


Fig. 4. Computational burden of the discussed schemes

bias of value unity yielding a $(r_i + 1) \times 1$ -dimensional excitation to the controller. The reason for such an augmentation is twofold:

i) If the denominator of (11) were considered, without such an augmentation the derivative would tend to infinity as the error vector moves towards the origin. However, having such a tendency in the adjustable controller parameters cannot result in convergence. When (14) is considered from the same point of view, together with the open form of matrix inversion, one sees that a convergent behavior enforces the tuning mechanism to behave like gradient descent. Although gradient descent can appropriately be used for controller training purposes, the structural simplicity of ADALINE will not allow the observation of a convergent behavior. This particular structure corresponds to linear time varying state feedback, which is well developed especially for systems whose dynamic representations are known totally or partly with known uncertainty bounds.

ii) When the sliding mode starts, the error vector rapidly converges to origin and the system starts tracking the desired trajectory precisely. However, since the magnitudes of the entries of the error vector are very close to zero, the corresponding controller parameters do not receive sufficient excitation to maintain the synthesis of target control sequence. In implementing RBFNN,

SFS and ANFIS architectures, the designer will not need such an augmentation since the parameter vector is persistently excited by the hidden neuron outputs of RBFNN or rule outputs of FL based structures.

A last remark here is on the applicability of FNN structure, to which solely the method in subsection 3.6 is applicable among what we have discussed.

4 An Illustrative Example

This section demonstrates the performance of the algorithm discussed in subsection 3.4 for a third order system studied previously by Roy et al [54] and Yilmaz et al [55]. The dynamic equation describing the system is given in (22).

$$\begin{aligned} \theta^{(3)} = & -0.5\theta - 0.5\dot{\theta}^3 - 0.5\ddot{\theta} \left| \dot{\theta} \right| + \left(1 + 0.1 \sin \left(\frac{\pi t}{3} \right) \right) \tau + \kappa_1(t) + \kappa_2(t) + \\ & (-0.05 + 0.25 \sin(5\pi t)) \theta + (-0.03 + 0.3 \cos(5\pi t)) \dot{\theta}^3 + \\ & (-0.05 + 0.25 \sin(7\pi t)) \ddot{\theta} \left| \dot{\theta} \right| \end{aligned} \quad (22)$$

where $\kappa_1(t) = 0.2 \sin(4\pi t)$ is the disturbance used in [54-55], and $\kappa_2(t)$ is the zero mean Gaussian noise corrupting the state information to be used by the controller additively. The work presented by Roy et al assume that the nominal system dynamics is known and the uncertain part is comprised of what we give as the last three terms in (22). The primary difference between what has been discussed so far and what is assumed in [54] should be stressed as the approaches we discuss only assume the achievability of the VSS task, hence the uncertainties are represented in the system dynamics, whose form is known but the details are not. As the controller, a three input single output ANFIS structure is used and the tuning is performed only on the defuzzifier parameters, which are initially set to zero. The rule base has 27 rules quantifying the relevant input variable as Negative, Zero or Positive. Once the rule outputs are evaluated, the crisp decision of the controller is computed as described in (1).

Parallel to [54], the reference state trajectory, which is described as $\theta_d = 0.5 \cos(\pi t/5)$ is used in the simulations. Initially, the states of the system have the following values, $\theta(0) = 1$, $\dot{\theta}(0) = 1$ and $\ddot{\theta}(0) = 1$. One important note here should be on the selection of λ . The value is taken as 5 in [54]; however we use $\lambda = 1$, because the behavior with this value results in a better system response. Figure 5 illustrates the trajectory followed in the phase space. The error vector hits the sliding surface several times and starts moving on it as enforced by the algorithm.

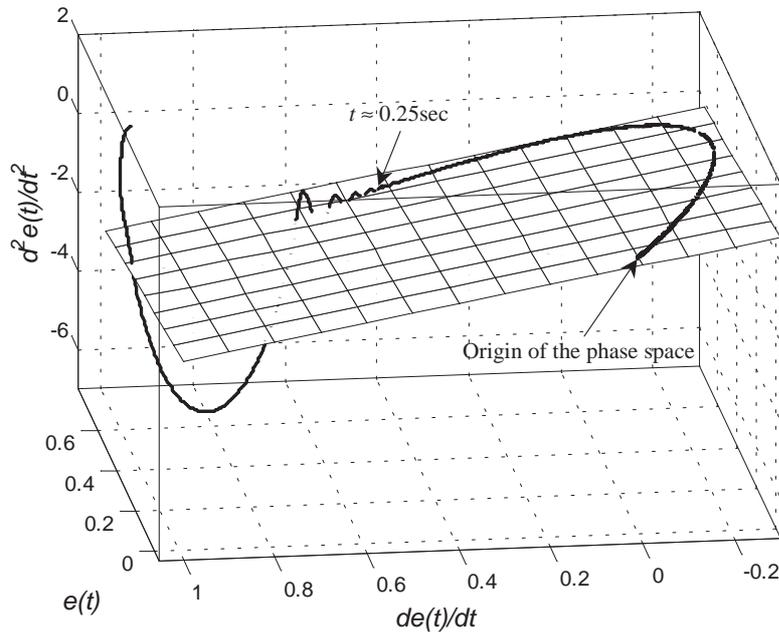


Fig. 5. Behavior in the phase space

5 Computational Intelligence in Variable Structure Control

What we have focused on so far mainly contemplates the use of VSS theory for parameter adaptation in CI. However, the integration of VSS technique with the architectural and algorithmic methods of CI can also be utilized in

- Chattering elimination through filtering [48,56]
- Design of the parameters of a conventional sliding mode controller [48,57-58]
- Modeling of the uncertainties [48,59-61]
- Generating a complementary control action [48,62-63]
- Generating the equivalent control and corrective control actions separately [48,64].

The use of CI in VSS may be a remedy in the situations where the available knowledge is insufficient to produce a safe control action. The selection of the uncertainty bound in this respect constitutes an apparent example. As the value of the uncertainty bound increases, the produced control action is more likely to have high frequency components having high magnitude, which arise through the sign measurement during the sliding mode. In such a situation, CI supported schemes can offer smoothed control signals with a

reasonable uncertainty bound selection. Furthermore, the conventional framework may underestimate the actuation speed of the system under control and may lead to unnecessarily large control inputs. In these cases, the tuning of VSS parameters, e.g. the slope of the sliding line, can be designed using the methods of CI. Being not limited to these, the methods of CI can be used as auxiliary subsystems for improving the control signal, i.e. a complementary control signal is produced so that the undesired effects of conventional sliding mode controller can be reduced. Last but not the least, the components of the control signal driving the system behavior to a predefined sliding regime can separately be realized by a learning system. This can eventually result in a comprehensible way of formulating the equivalent control and the corrective control.

6 Conclusion

This study discusses the design of a VSS theory based training strategies for CI, when the traditional gradient based training approaches are utilized for which, some handicaps arise due to the imperfect modeling, noisy observations or time varying parameters. If the effects of these factors are transformed to the cost hypersurface, whose dimensionality is determined by the adjustable design parameters, it becomes evident that the surface may have directions along which the sensitivity derivatives assume large values. In these cases, gradient based optimization procedures tend to evaluate large parametric displacements, which can eventually lead to a locally divergent behavior. In control engineering practice, such a behavior constitutes a potential danger from a safety point of view. The approaches presented in this work take care of the mentioned difficulties. Since the VSS theory is well known with its robustness property, a training strategy equipped with which retains a high degree of robustness against disturbances and uncertainties. When these approaches are considered for the training of intelligent controllers, under the assumption that the VSS task is achievable, the task is fulfilled without knowing the analytic details describing the plant dynamics. In order to corroborate the performance claims, tracking control of a third order nonlinear system is presented. The behavior in the phase space clearly demonstrates the superior performance despite the unavailability of system-specific details.

References

1. Jang J.-S.R., Sun C.-T., Mizutani E. (1997) *Neuro-Fuzzy and Soft Computing*, PTR Prentice-Hall
2. Berenji H. (1992) *Fuzzy and Neural Control*, In: P. J. Antsaklis, and K. M. Passino (Eds.), *An Introduction to Intelligent and Autonomous Control*, 215-236, Kluwer Academic Publishers, The Netherlands
3. Hornik K. (1989) Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, **2**, 359-366

4. Funahashi, K. (1989) On the Approximate Realization of Continuous Mappings by Neural Networks, *Neural Networks*, **2**, 183-192
5. Cybenko G. (1989) Approximation by Superpositions of a Sigmoidal Function, *Mathematics of Control, Signals, and Systems*, **2**, 303-314
6. Haykin S. (1994) *Neural Networks*, Macmillan College Printing Company, New Jersey
7. Gupta M.M., Rao D.H. (1993) Dynamical Neural Units with Applications to the Control of Unknown Nonlinear Systems, *Journal of Intelligent and Fuzzy Systems*, **1**, 1, 73-92
8. Wang Y-J., Lin C-T. (1998) Runge-Kutta Neural Network for Identification of Dynamical Systems in High Accuracy, *IEEE Transactions on Neural Networks*, **9**, 2, 294-307, March
9. Zadeh L.A. (1965) Fuzzy Sets, *Information and Control*, **8**, 338-353
10. Wang L. (1997) *A Course in Fuzzy Systems and Control*, PTR Prentice-Hall
11. Takagi T., and Sugeno M. (1985) Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Transactions on Systems, Man, and Cybernetics*, **15**, 1, 116-132, January
12. Rumelhart D.E., Hinton G.E., and Williams R.J. (1986) Learning Internal Representations by Error Propagation, in D. E. Rumelhart and J. L. McClelland, (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, **1**, 318-362, MIT Press, Cambridge, M.A.
13. Hagan M. T., M. B. Menhaj (1994) Training Feedforward Networks with the Marquardt Algorithm, *IEEE Transactions on Neural Networks*, **5**, 6, 989-993, November
14. Emelyanov S. V. (1959) Control of First Order Delay Systems by means of an Astatic Controller and Nonlinear Correction, *Automat. Remote Contr.*, **8**, 983-991
15. Utkin V. I. (1977) Variable Structure Systems with Sliding Modes, *IEEE Transactions Automatic Control*, **22**, 212-222
16. Young K. D., Utkin V. I., Ozguner U. (1999) A Control Engineer's Guide to Sliding Mode Control, *IEEE Transactions on Control Systems Technology*, **7**, 3, 328-342, May
17. Hung J.Y., Gao W., Hung J. C. (1993) Variable Structure Control: A survey, *IEEE Transactions on Industrial Electronics*, **40**, 1, 1-9, February
18. Zinober A.S.I. (1994) (Ed), *Variable Structure and Lyapunov Control*, Springer-Verlag
19. Young K.K. (1993) (Ed), *Variable Structure Control for Robotics and Aerospace Systems*, Elsevier-Science
20. Utkin V.I. (1992) *Sliding Modes in Control Optimization*, Springer Verlag, New York
21. Young K.D. (1978) Controller Design for a Manipulator Using Theory of Variable Structure Systems, *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-8**, 210-218
22. Slotine J.J.E., Sastry S.S. (1983) Tracking Control of Nonlinear Systems Using Sliding Surfaces with Application to Robot Manipulators, *International Journal of Control*, **38**, 465-492
23. Hashimoto H., Maruyama K., Harashima F. (1987) A Microprocessor Based Robot Manipulator Control with Sliding Mode, *IEEE Transactions on Industrial Electronics*, **34**, 11-18

24. Wijesoma S.W. (1990) Robust Trajectory Following of Robots Using Computed Torque Structure with VSS, *International Journal of Control*, **52**, 935-962
25. Denker A., Kaynak O. (1994) Application of VSC in Motion Control Systems, in *Variable Structure and Lyapunov Control*, A.S.I. Zinober (Ed.), Springer-Verlag, Chapter 17, 365-383
26. Kaynak O., Harashima F., Hashimoto H. (1984) Variable Structure Systems Theory, as applied to Sub-time Optimal Position Control with an Invariant Trajectory, *Trans. IEE of Japan, Sec. E*, **104**, 47-52
27. Bartoszewicz A. (1988) On the Robustness of Variable Structure Systems in the Presence of Measurement Noise, *Proc. IEEE Industrial Electronics Society Annual Conference, IECON'99, Aachen, Germany*, 1733-1736, Aug. 31-Sept. 4
28. Slotine J.J.E., Li W. (1991) *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice-Hall
29. Elmali, H., Olgac N. (1992) Robust Output Tracking Control of Nonlinear MIMO Systems via Sliding Mode Technique, *Automatica*, **28**, 145-151
30. Izosimov D. B., Utkin V. I. (1981) On Equivalence of Systems with Large Coefficients and Systems with Nonlinear Control, *Automation and Remote Control*, **11**, 189-191
31. Tunay I., Kaynak O. (1996) Provident Control of an Electrohydraulic Servo with Experimental Results, *Mechatronics*, **6**, 3, 249-260
32. Tunay I., Kaynak O. (1995) A New Variable Structure Controller for Affine Nonlinear Systems with Non-matching Uncertainties, *International Journal of Control*, **62**, 4, 917-939
33. Yen J., Langari R. (1999) *Fuzzy Logic*, PTR Prentice-Hall, New Jersey
34. Passino K.M., Yurkovich S. (1998) *Fuzzy Control*, Addison-Wesley, California
35. Wang L.-X. (1994) *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*, PTR Prentice-Hall
36. Efe M.O., Kaynak O., Wilamowski B.M. (2000) Creating a Sliding Mode in a Motion Control System by Adopting a Dynamic Defuzzification Strategy in an Adaptive Neuro Fuzzy Inference System, *Proc. IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation, IECON-2000, Nagoya, Japan*, 894-899, Oct. 22-28
37. Efe M.O., Kaynak O. (1999) A Comparative Study of Neural Network Structures in Identification of Nonlinear Systems, *Mechatronics*, **9**, 3, 287-300
38. Sanner R.N., Slotine J.J.E. (1992) Gaussian Networks for Direct Adaptive Control, *IEEE Transactions on Neural Networks*, **3**, 6, 837-863
39. Sira-Ramirez H., Colina-Morles E. (1995) A Sliding Mode Strategy for Adaptive Learning in Adalines, *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, **42**, 12, 1001-1012, December
40. Hsu L., Real J.A. (1997) Dual Mode Adaptive Control Using Gaussian Neural Networks, *Proc. of the 36th Conference on Decision and Control, (CDC), New Orleans, LA*, 4032-4037
41. Hsu L., Real J.A. (1999) Dual Mode Adaptive Control, *Proc. of the IFAC'99 World Congress, Beijing*, **K**, 333-337
42. Yu X., Zhihong M., Rahman S.M.M. (1998) Adaptive Sliding Mode Approach for Learning in a Feedforward Neural Network, *Neural Computing & Applications*, **7**, 289-294
43. Parma G.G., Menezes B.R., Braga A.P. (1998) Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks, *Electronics Letters*, **34**, 1, 97-98, January

44. Sira-Ramirez H., Colina-Morles E., Rivas-Echeverria F. (2000) Sliding Mode-Based Adaptive Learning in Dynamical-Filter-Weights Neurons, *International Journal of Control*, **73**, 8, 678-685
45. Efe M.O., Kaynak O., Yu X. (2000) Sliding Mode Control of a Three Degrees of Freedom Anthropoid Robot by Driving the Controller Parameters to an Equivalent Regime, *Transactions of the ASME: Journal of Dynamic Systems, Measurement and Control*, **122**, 4, 632-640, December
46. Efe M.O., Kaynak O. (2000) On Stabilization of Gradient Based Training Strategies for Computationally Intelligent Systems, *IEEE Transactions on Fuzzy Systems*, **8**, 5, 564-575, October
47. Efe M.O., Kaynak O., Wilamowski B.M. (2000) Stable Training of Computationally Intelligent Systems By Using Variable Structure Systems Technique, *IEEE Transactions on Industrial Electronics*, **47**, 2, 487-496, April
48. Kaynak O., Erbatır K., Ertugrul M. (2001) The Fusion of Computationally Intelligent Methodologies and Sliding-Mode Control - A Survey, *IEEE Transactions on Industrial Electronics*, **48**, 1, 4-17, February
49. Efe M.O. (2000) Variable Structure Systems Theory Based Training Strategies for Computationally Intelligent Systems, Ph.D. Dissertation, Bogazici University
50. Yu X. Efe M.O., Kaynak O. (2001) A Backpropagation Learning Framework for Feedforward Neural Networks, in *Proc. of the 2001 IEEE Int. Symposium on Circuits and Systems (ISCAS'01)*, **III**, pp. 700-702, May 6-9, Sydney, Aust
51. Zhao Y. (1996) On-line Neural Network Learning Algorithm with Exponential Convergence Rate, *Electronic Letters*, **32**, 15, 1381-1382, July
52. Bersini H., Gorrini V. (1997) A Simplification of the Backpropagation Through Time Algorithm for Optimal Neurocontroller, *IEEE Transactions Neural Networks*, **8**, 2, 437-441, March
53. Erbatır K., Kaynak O., Sabanovic A. (1999) A Study on Robustness Property of Sliding Mode Controllers: A Novel Design and Experimental Investigations, *IEEE Transactions on Industrial Electronics*, **46**, 5, 1012-1018
54. Roy R.G., Olgac N. (1997) Robust Nonlinear Control via Moving Sliding Surfaces - n-th Order Case, *Proc. of the 36th Conference on Decision and Control*, San Diego, California, U.S.A., December 943-948
55. Yilmaz C., Hurmuzlu Y. (2000) Eliminating the Reaching Phase from Variable Structure Control, *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, **122**, 4, 753-757, December
56. Hwang Y.R., Tomizuka M. (1994) Fuzzy Smoothing Algorithms for Variable Structure Systems, *IEEE Transactions on Fuzzy Systems*, **2**, 4, 277-284
57. Choi S.B., Kim M.S. (1997) New Discrete-Time Fuzzy-Sliding-Mode Control with Application to Smart Structures, *Journal of Guidance Control and Dynamics*, **20**, 5, 857-864
58. Erbatır K., Kaynak O., A. Sabanovic (1996) I. Rudas, Fuzzy Adaptive Sliding Mode Control of a Direct Drive Robot, *Robotics and Autonomous Systems*, **19**, 2, 215-227
59. Chen C.S., Chen W.L. (1998) Robust Adaptive Sliding-Mode Control Using Fuzzy Modeling for an Inverted-Pendulum System, *IEEE Transactions on Industrial Electronics*, **45**, 2, 297-306
60. Yu X., Man Z.H., Wu B.L. (1998) Design of Fuzzy Sliding-Mode Control Systems, *Fuzzy Sets and Systems*, **95**, 3, 295-306

61. Yoo B., Ham W. (1998) Adaptive Fuzzy Sliding Mode Control of Nonlinear System, *IEEE Transactions on Fuzzy Systems*, **6**, 2, 315-321
62. Ha Q.P. (1996) Robust Sliding Mode Controller with Fuzzy Tuning, *Electronics Letters*, **32**, 17, 1626-1628
63. Ha Q.P. (1997) Sliding Performance Enhancement with Fuzzy Tuning, *Electronics Letters*, **33**, 16, 1421-1423
64. Ertugrul M., Kaynak O. (2000) Neuro Sliding Mode Control of Robotic Manipulators, *Mechatronics*, **10**, 1-2, 243-267