

A Lyapunov Based Stable Online Learning Algorithm For Nonlinear Dynamical Systems Using Extreme Learning Machines

Vijay Manikandan Janakiraman*, XuanLong Nguyen[†], Dennis Assanis[‡]

*Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA,

Email: vijai@umich.edu

[†]Department of Statistics, University of Michigan, Ann Arbor, MI, USA,

Email: xuanlong@umich.edu

[‡]Stony Brook University, NY, USA,

Email: dennis.assanis@stonybrook.edu

Abstract— Extreme Learning Machine (ELM) is a promising learning scheme for nonlinear classification and regression problems and has shown its effectiveness in the machine learning literature. ELM represents a class of generalized single hidden layer feed-forward networks (SLFNs) whose hidden layer parameters are assigned randomly resulting in an extremely fast learning speed along with superior generalization performance. It is well known that the online sequential learning algorithm (OS-ELM) based on recursive least squares [1] might result in ill-conditioning of the Hessian matrix and hence instability in the parameter estimation. To address this issue, the stability theory of Lyapunov is utilized to develop an online learning algorithm for temporal data from dynamic systems and time series. The developed algorithm results in parameter estimation that is globally asymptotically stable. Simulations results of the developed algorithm compared against online sequential ELM (OS-ELM) and the offline batch learning ELM (O-ELM) show that the online Lyapunov ELM algorithm can perform online learning at reduced computation, comparable accuracy and with a guarantee on the boundedness of the estimated system.

I. INTRODUCTION

DYNAMIC SYSTEMS are systems that evolve over time. The states (or solutions) of a dynamic system evolve as a function of their past states and external excitations. The dynamic systems that depend only on the past states without external excitations are categorized as time-series. Dynamic systems encapsulates a large class of real-world systems [2] including mechanical, electrical, chemical and biological to name a few. Data mining on dynamic systems has been as significant as on static systems but often the time connection in a dynamic system is usually not utilized. For instance, neural networks [3], [4] and support vector machines [5] have been used in modeling dynamic systems but algorithms designed for static data with an i.i.d assumption (data sampled from an independent and identical distribution) are used. The temporal aspects of the data useful for parameter estimation and decision making for dynamical systems are typically not taken into account.

Modeling of dynamic systems using data is often referred to as system identification in the controls literature [3] and is

a significant area of research solving problems in modeling [3], [4], fault detection, optimization and control [6], [7]. In most cases, an online regression algorithm updates the parameters by minimizing a given cost function and decisions are made simultaneously based on the estimated model (as in case of adaptive control for instance [7]). In such situations, it is vital for the parameter estimation to be stable so that model based decisions are valid. Hence a guarantee of stability and boundedness is of extreme importance.

Application of Lyapunov stability theory for model identification is common in controls literature [8] but less common in machine learning. For instance, a Lyapunov approach has been used for identification using radial basis function neural networks [9] and GLO-MAP models [10] were among the few. The parameter update in such methods involved complex gradient calculation in real time or first estimating a linear model and then estimating a nonlinear difference using orthonormal polynomial basis functions. In this paper, the recently developed random projection based neural networks (ELM) has been used as the underlying model structure owing to its superior properties in general nonlinear regression problems. In spite of its known advantages (see section II), an over-parameterized ELM suffers from the ill-conditioning problem when recursive least squares type update is performed (as in OS-ELM). This results in poor regularization behavior [11], [12], [13], [14], and more importantly, leads to an unbounded growth of the model parameters and predictions. The goal of this paper is to address the above issue by developing a stable online learning algorithm for ELM using Lyapunov approach which guarantees the boundedness of the model parameters as well as predictions even when the model is over-parameterized.

The paper is organized as follows. The basics of ELM modeling is briefed in section II. The proposed online learning algorithm is derived in section III while simulation results and qualitative evaluations are discussed in section IV.

II. EXTREME LEARNING MACHINES - BACKGROUND

Extreme Learning Machine (ELM) is an emerging learning paradigm for multi-class classification and regression problems [15], [16]. The highlight of ELM compared to the

This material is based upon work supported by the Department of Energy [National Energy Technology Laboratory] under Award Number(s) DE-EE0003533.

other state of the art methodologies like neural networks, support vector machines is that the training speed of ELM is extremely fast. The key enabler for ELM's training speed is the random assignment of input layer parameters which do not require computationally intensive tuning to the data. In such a setup, the output layer parameters can be determined analytically using least squares. Some of the attractive features of ELM [15] are listed below

- 1) ELM is an universal approximator (suit a wide class of systems)
- 2) ELM results in the smallest training error without getting trapped in local minima (better accuracy)
- 3) ELM does not require iterative training (faster training)
- 4) ELM solution has the smallest norm of weights (better generalization).

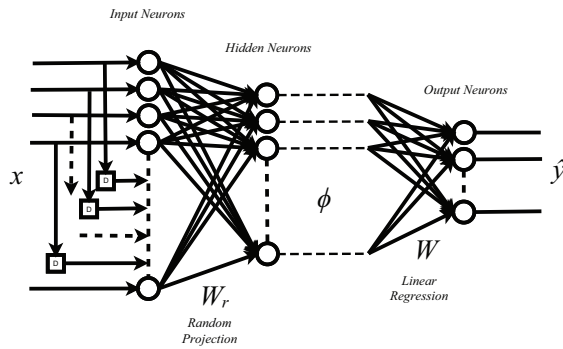


Fig. 1. ELM Model Structure.

ELM is developed from a machine learning perspective and typically data observations are considered independent and identically distributed. As a result, application of ELM to a dynamic system may not be direct as the data is connected in time. However, ELM can be applied for system identification in discrete time by using a series-parallel formulation [3]. A generic nonlinear identification using the nonlinear auto regressive model with exogenous input (NARX) is considered as follows

$$y(k) = f[u(k-1), \dots, u(k-n_u), y(k-1), \dots, y(k-n_y)] \quad (1)$$

where $u(k) \in \mathbb{R}^{u_d}$ and $y(k) \in \mathbb{R}^{y_d}$ represent the inputs and outputs of the system respectively, k represents the discrete time index, $f(\cdot)$ represents the nonlinear function mapping specified by the model, n_u, n_y represent the number of past input and output samples required (order of the system) for prediction while u_d and y_d represent the dimension of inputs and outputs respectively.

A. Offline ELM learning algorithm (O-ELM)

The input-output measurement sequence of system (1) can be converted to the form of training data as required by ELM

$$\{(x_1, y_1), \dots, (x_N, y_N)\} \in (\mathcal{X}, \mathcal{Y}) \quad (2)$$

where \mathcal{X} denotes the space of the input features (Here $\mathcal{X} = \mathbb{R}^{u_d n_u + y_d n_y}$ and $\mathcal{Y} = \mathbb{R}^{y_d}$) and x represent the augmented input vector obtained by appending the input and output measurements from the system as given by the following equation

$$x = [u(k-1), \dots, u(k-n_u), y(k-1), \dots, y(k-n_y)]^T. \quad (3)$$

The ELM is an unified representation of single layer feed-forward networks (SLFN) whose structure can be seen in Fig. 1. The system inputs u and outputs y at time instant k along with their histories (indicated by signals delayed using unit delays D) are given as inputs x to the model. The inputs are randomly projected to a high dimension feature space using random matrix W_r . The model prediction \hat{y} is given by

$$\hat{y} = \hat{W}^T [g(W_r^T x + b_r)] \quad (4)$$

where g represents the hidden layer activation function and W_r, W represents the input and output layer parameters respectively (\hat{W} represents estimated values). The elements of W_r and b_r can be assigned based on any continuous random distribution [16] and remains fixed during the learning process. The number of hidden neurons (n_h) determine the dimension of the transformed feature space and the hidden layer is equipped with a nonlinear activation function similar to the traditional neural network architecture. It should be noted that nonlinear regression using neural networks for instance, the input layer parameters W_r and W are simultaneously adjusted during training. Since there is a nonlinear connection between the two layers, iterative techniques are typically used. ELM, however, avoids the iterative training as the input layer parameters are randomly selected [15]. Hence, the training step of ELM reduces to finding a least squares solution to the output layer parameters W given by

$$\min_W \{ \|HW - Y\|^2 + \lambda \|W\|^2 \} \quad (5)$$

$$\hat{W} = (H^T H + \lambda I)^{-1} H^T Y \quad (6)$$

where λ represents the regularization coefficient that determines a tradeoff between minimizing the training error and maintaining model simplicity, Y represents the vector of outputs or targets and $\phi = H^T = g(W_r^T x + b_r)$ the hidden layer output matrix as termed in literature.

B. Online ELM learning algorithm (OS-ELM)

In the batch training mode (offline training), all the data are assumed to be present. However, for an online system identification problem, data is sampled continuously and is available one by one. This requires that the sequential learning algorithm be modified to perform online learning as follows (see also [1]).

As an initialization step, a set of data observations (N_0) are required to initialize the H_0 and W_0 by solving the following optimization problem

$$\min_{W_0} \{ \|H_0 W_0 - Y_0\|^2 + \lambda \|W_0\|^2 \} \quad (7)$$

$$H_0 = [g(W_r^T x_0 + b_r)]^T \in \mathbb{R}^{N_0 \times n_h}. \quad (8)$$

The solution W_0 is given by

$$W_0 = K_0^{-1} H_0^T Y_0 \quad (9)$$

where $K_0 = H_0^T H_0$. Suppose given another new data x_1 , the problem becomes

$$\min_{W_1} \left\| \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} W_1 - \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} \right\|^2. \quad (10)$$

The solution can be derived as

$$\begin{aligned} W_1 &= W_0 + K_1^{-1} H_1^T (Y_1 - H_1 W_0) \\ K_1 &= K_0 + H_1^T H_1. \end{aligned}$$

Based on the above, a generalized recursive algorithm for updating the least-squares solution can be computed as follows

$$\begin{aligned} M_{k+1} &= M_k - M_k H_{k+1}^T (I + H_{k+1} M_k H_{k+1}^T)^{-1} H_{k+1} M_k \\ &\quad (11) \\ W_{k+1} &= W_k + M_{k+1} H_{k+1}^T (Y_{k+1} - H_{k+1} W_k) \quad (12) \end{aligned}$$

where M represents the covariance of the parameter estimate.

III. LYAPUNOV BASED PARAMETER UPDATE ALGORITHM

A general nonlinear discrete time dynamic system model representing the underlying phenomena of interest can be given by

$$z(k+1) = f(z(k), u(k)) \quad (13)$$

where $z(k)$ and $u(k)$ represent the system states and inputs. The system in (13) is assumed to satisfy the following assumptions in order for the given nonlinear system to be identified via a Lyapunov function [17].

- 1) The system is completely controllable and completely observable.
- 2) The function f is continuously differentiable and satisfies $f(\mathbf{0}) = 0$. The function also satisfies global Lipschitz's condition [18] which guarantees existence and uniqueness of solution to the differential equation (13).
- 3) The function f is stationary; i.e., $f(\cdot)$ does not depend explicitly on time.

Now, the model in (13) can be expressed as

$$\begin{aligned} z(k+1) &= Az(k) + g(z(k), u(k)) \quad (14) \\ &= Az(k) + W_*^T \phi(k) + \epsilon. \quad (15) \end{aligned}$$

In the above expression, $z \in \mathbb{R}^n$, n representing the number of states, $g(z(k), u(k)) = f(z(k), u(k)) - Az(k)$ represents the system nonlinearity and assuming ELM can model $g(z(k), u(k))$ with an accuracy of ϵ using parameters W_* . The above assumption is valid in theory as ELM is an universal approximator [15]. The matrix A is included so as to maintain asymptotic stability of equation (15) and can be chosen as any matrix with eigen values in the unit circle

[2]. The parametric model of the system can be constructed assuming the same structure as equation (15) by

$$\hat{z}(k+1) = A\hat{z}(k) + \hat{W}^T(k)\phi(k) \quad (16)$$

where $\hat{W}(k)$ represents the parameter estimate of W_* at time index k . The state error is given by the following

$$\begin{aligned} e(k+1) &= z(k+1) - \hat{z}(k+1) \\ &= Ae(k) + \tilde{W}^T(k)\phi(k) + \epsilon \quad (17) \end{aligned}$$

where $\tilde{W}(k)$ represents the parameter error ($\tilde{W}(k) = W_* - \hat{W}(k)$). It should be noted that $\phi(k)$ is common to both system model and the parametric model which means the inputs and states of the plant are fed to the model indicating a series-parallel architecture [19].

A positive definite, decrescent and radially unbounded [8] function $V(e, \tilde{W}, k)$ (18) is used to construct the parameter update law given by equation in equation (19) (See Appendix for detailed derivations). Lyapunov functions are used to establish stability of nonlinear systems and a thorough mathematical treatment of Lyapunov based system identification can be found in [17]. Here Γ represents the gain of learning and P is a matrix solution of the Lyapunov equation $A^T P A - P = -Q$ where $Q \in \mathbb{R}^{n \times n}$ is a positive definite matrix defined by the user.

$$V(e, \tilde{W}, k) = e^T(k) P e(k) + \frac{1}{2} \text{tr}(\tilde{W}(k) \Gamma \tilde{W}^T(k)) \quad (18)$$

$$\begin{aligned} \hat{W}(k+1) &= \\ &= \hat{W}(k) + \Gamma^{-T} \phi(k) [e(k+1) + Ae(k)]^T P \quad (19) \end{aligned}$$

The Lyapunov based online learning algorithm is given by (19). Now, applying the update law, the change in Lyapunov function becomes

$$\begin{aligned} \Delta V(k) &= -e^T(k) Q e(k) + e^T(k) A^T P \epsilon + \epsilon^T P e(k+1) \\ &= -e^T(k) Q e(k) + e^T(k) A^T P \epsilon + \epsilon^T P (e(k) + \Delta e(k)) \\ &= -e^T(k) Q e(k) + e^T(k) A^T P \epsilon + \epsilon^T P e(k) \\ &\leq -|\lambda_{\max} Q| \|e(k)\|^2 + \|\epsilon\| \|P\| \|e(k)\| + \|\epsilon\| \|P\| \|A\| \|e(k)\| \\ &< 0 \text{ if } \|e(k)\| \geq \frac{\|\epsilon\| \|P\| \|I + A\|}{|\lambda_{\max} Q|} = E_\epsilon. \quad (20) \end{aligned}$$

When the above condition is satisfied, equation (20) can be written as

$$\Delta V(k) \leq -|\lambda_{\max} Q| \|e(k)\|^2 < 0 \quad (21)$$

For all $k > 0$, $V(k) - V(0) < 0$ so that $V(k) < V(0)$. Also, from the positive definiteness of V , $V(e, \tilde{W}, k) > 0$ for all $[e, \tilde{W}, k]^T > 0$. This implies that $0 < V(k) < V(0)$ for all $k > 0$. Hence $V(k) \in \mathbf{L}_\infty$, i.e.,

$$\lim_{k \rightarrow \infty} V(k) = V_\infty < \infty. \quad (22)$$

Also, $V(k)$ is a function of $e(k)$ which results in $e(k) \in \mathbf{L}_\infty$. This implies $\hat{z}(k) \in \mathbf{L}_\infty$ assuming the states of the plant $z(k)$ are bounded. Hence the states of the estimation model

$\hat{z}(k)$ do not blow up eliminating possible errors during model based decision making. Also, from equation (21),

$$|\lambda_{max}Q|||e(k)||^2 \leq -\Delta V(k) = V(0) - V_\infty \quad (23)$$

which implies $e(k) \in \mathbf{L}_2$. Combining the results, $e(k) \in \mathbf{L}_2 \cap \mathbf{L}_\infty$. Using the special case of Barbalat's Lemma [8], it can be shown that $e(k) \rightarrow 0$ as $k \rightarrow \infty$. Hence the error between the true model and the estimation model converges to zero and the estimation model becomes a one-step ahead predictive model of the nonlinear system.

It should be noted that the parameter matrix \hat{W} converges to the true parameters W_* only under conditions of persistence of excitation [8]. If the persistent excitation condition is not met, \hat{W} converges to some constant values. For general nonlinear systems where persistence of excitation may not be met [20], a multi-step signal varying between extreme values of the inputs are typically used [4], [5]. For any given step input, the update law in (19) asymptotically reduces the error $e(k)$ to zero. Intuitively, the algorithm leads to a local parameter convergence (depending on the available excitation strength) to its estimates and as the system is excited more and more, the estimates moves towards the true values. This can be observed in simulations as well.

IV. SIMULATION RESULTS

In this section, the developed algorithm is applied to two simple nonlinear dynamic systems and its working analyzed. The performance of the Lyapunov based algorithm (L-ELM) is compared against the baseline nonlinear OS-ELM algorithm. The results of linear identification models based on recursive least squares (RLS) and linear Lyapunov based method (LLin) are also compared to demonstrate that the chosen examples are truly nonlinear. Also, the results of an offline batch learning ELM algorithm (O-ELM) is included to evaluate if the L-ELM trades off accuracy for computational simplicity.

The performance metrics used are one-step ahead prediction (OSAP) and multi-step ahead prediction (MSAP) root mean squared errors (RMSE). For MSAP, the models are converted from a series-parallel architecture to a parallel architecture [19] where the predictions of the model is fed back as inputs along with external excitation creating a recursive prediction over the prediction horizon. In this way, the evaluation of the model being close to the actual dynamic system can be performed. Such a metric is important in model based decision making applications [3], [4].

A. Example - I: Simple Scalar System

Consider the following single input single output (SISO) system

$$\begin{aligned} z(k+1) &= \sin\left(\frac{z(k)^3}{10}\right) + \frac{u(k)^2}{10} \\ z(0) &= 0. \end{aligned} \quad (24)$$

Here the augmented vector becomes $x(k) = [z(k), u(k)]^T$. The inputs vary between -1 and +1 in an uniformly distributed multi step pseudorandom pattern to get information

about majority of the operating regions of the system. The parameters of identification are chosen as follows: $A = 0.1, n_h = 5, Q = 1$ and $\Gamma = 10I_{5,5}$.

It can be seen from Fig. 2 that the parameters of the L-ELM appear to be converging to some steady state values which indicates that the nonlinear system is close to being identified. However, the OS-ELM parameters grow in an unstable manner and then converge back. Even though the parameters converge, the instability can be observed. This will be discussed more in section IV-C. The linear models do not have sufficient degrees of freedom and oscillates without converging (The oscillations are not seen because the time scale is very large, please refer to Fig. 5).

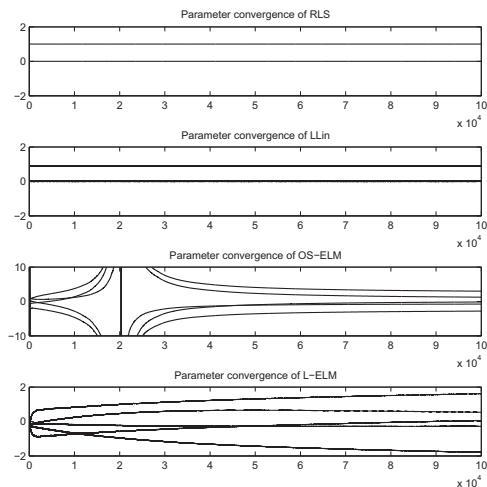


Fig. 2. Comparison of parameter evolution for the L-ELM with OS-ELM and the linear models for the simple scalar system

TABLE I
PERFORMANCE COMPARISON OF L-ELM WITH OS-ELM AND THE LINEAR MODELS FOR THE SIMPLE SCALAR SYSTEM

| | OSAP RMSE | MSAP RMSE | $\ W\ $ |
|--------|-----------|-----------|---------|
| RLS | 0.0098 | 0.0494 | 0.981 |
| Llin | 0.0099 | 0.0556 | 0.834 |
| OS-ELM | 0.0059 | 0.0008 | 5.915 |
| L-ELM | 0.0028 | 0.0012 | 2.412 |
| O-ELM | 0.0012 | 0.0007 | 3.655 |

The one-step ahead and multi-step ahead prediction performances of the models can be summarized in Fig. 3 and Fig. 4 respectively. The nonlinear models performed well in both one-step ahead and multi-step ahead predictions. The linear models are clearly unsuitable for MSAP and cannot be used for dynamic simulations. The high RMSE values indicate the inability of linear models to capture the system behavior with limited number of degrees of freedom. The multi-step ahead prediction is done by feeding back the model predictions along with an input of the form $u(k) = 0.2 * \sin(\frac{2\pi k}{50}) + 0.8 * \cos(\frac{2\pi k}{50})$ so that the input covers the region of operation of the models.

The prediction RMSE and the norm of the estimated parameters are listed in Table I. It can be seen that the L-ELM achieves a lower norm of parameters compared to the OS-ELM and O-ELM which can be attributed to the stable learning method which bounds the parameter growth. The results of the offline (batch learning) ELM model indicates that the performance of the developed online learning algorithm is comparable to that of batch learning.

It should be noted that the observed results and discussion are for a given random initialization of the hidden layer parameters (W_r, b_r), a given initial conditions of the estimated parameters (W_0) and a given set of design parameters of the algorithm (A, Γ, P). A different gain value (Γ) for instance might lead to a different convergence behavior and might result in different prediction errors and different norm of parameters.

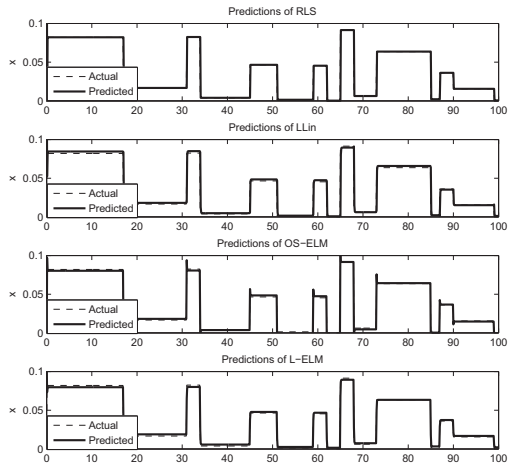


Fig. 3. Comparison of one-step ahead predictions for the L-ELM with OS-ELM and the linear models for the simple scalar system

B. Example - II: A more complex system

In this section, a more complex example of a system with two states and two inputs are considered as follows

$$x_1(k+1) = \sin\left(\frac{x_1(k)}{1+x_2^2(k)} + u_1(k)\right) \quad (25)$$

$$x_2(k+1) = \cos\left(1 - \frac{x_1(k)x_2(k)}{1+x_2^2(k)} - u_2(k)\right) \quad (26)$$

$$x(0) = [0, 0]^T \quad (27)$$

Similar to the previous case, the inputs vary between -1 and +1 independently in a multi-step pseudorandom pattern. The algorithm design parameters are chosen as follows: $A = 0.1I_{2,2}$, $n_h = 8$, $Q = I_{2,2}$ and $\Gamma = 10I_{8,8}$.

It can be seen from Fig. 5 that the parameters of the L-ELM do not converge completely indicating that either the number of hidden neurons is less or the identification time is insufficient to have sufficient excitations. Hence parameter oscillations are observed. Similarly, the parameters of the

LLin model also oscillates indicating that the linear models do not have enough degrees of freedom to capture the nonlinear behavior.

An important observation to be made is that the Lyapunov method (for linear or nonlinear model) solves the error dynamics differential equation (17) and finds the solution to the differential equation (19). The use of PRBS type of signal forces the update law to reduce the error and updates the parameters to only adapt to the system locally given by that particular excitation. If the excitation signal had several frequencies simultaneously (sum of sinusoids), the convergence would be global as in case of linear system identification [8]. However, for nonlinear systems, sufficient excitation is never completely achieved [8], [3] indicating that the parameters locally adapt to particular excitations and if carried out for an infinite time, might achieve global convergence. This is an existing problem to all types of nonlinear identification algorithms and is not addressed in this paper. Hence the reason for parameter oscillation for L-ELM for this example. A good trick is to slow down the local learning process using the gain parameter Γ so that aggressive local convergence is reduced and global convergence is made faster. This is being presently investigated by the authors.

TABLE II
PERFORMANCE COMPARISON OF L-ELM WITH OS-ELM AND THE LINEAR MODELS FOR THE MORE COMPLEX SYSTEM

| | OSAP RMSE | MSAP RMSE | $\ W\ $ |
|--------|-----------|-----------|---------|
| RLS | 0.147 | 0.67 | 0.92 |
| Llin | 0.128 | 0.69 | 0.83 |
| OS-ELM | 0.124 | 0.32 | 8.5 |
| L-ELM | 0.082 | 0.235 | 6.9 |
| O-ELM | 0.088 | 0.295 | 13.26 |

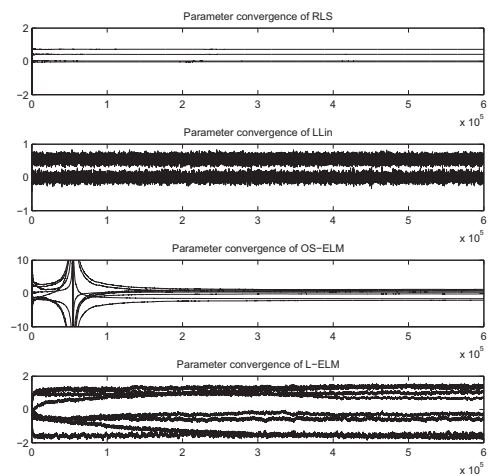


Fig. 5. Comparison of parameter evolution for the L-ELM with OS-ELM and the linear models for the more complex system

The prediction performance is summarized in Fig. 6 and Fig. 7 along with Table II. The linear models have high

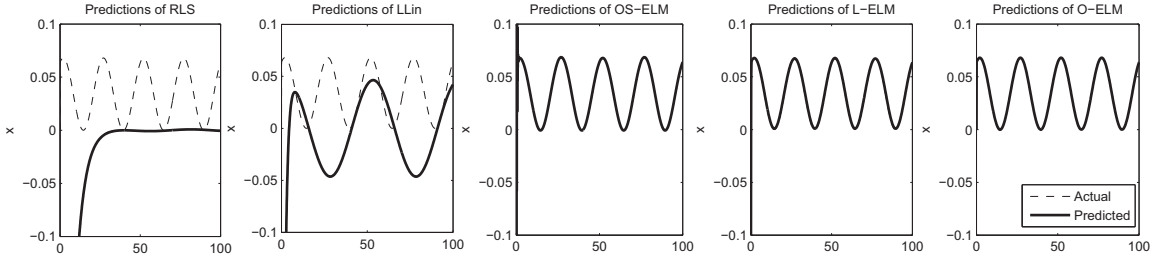


Fig. 4. Comparison of multi-step ahead predictions for the L-ELM with OS-ELM, O-ELM and the linear models for the simple scalar system

RMSE values and hence not suitable for identification of the given nonlinear system. The multi-step ahead prediction is done by feeding back the model's predictions along with external inputs of the form $u(k) = [\sin(\frac{2\pi k}{10}), \cos(\frac{2\pi k}{10})]^T$. The results of the offline ELM model is included to show that the performance of the developed algorithm is comparable to the batch learning model indicating that there is no compromise in accuracy by performing online learning using the Lyapunov method.

It can be seen from Table II that the norm of estimated parameters are smaller for the L-ELM indicating better generalization compared to OS-ELM and O-ELM. It should be noted that the offline models are developed and validated using the same data set (sub-sampled to reduce computation) on which the online learning was performed. In this example, the MSAP RMSE is lower for the L-ELM compared to the OS-ELM while in the previous example, the MSAP RMSE for OS-ELM was better indicating that a conclusion cannot be made as which method outperforms the other in terms of prediction accuracy. However, the prime advantage of the L-ELM comes from its stable parameter evolution which is briefed in the following subsection.

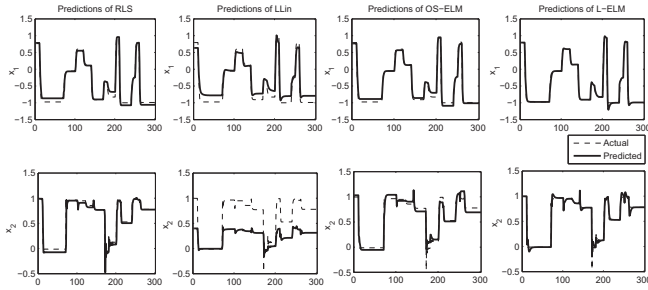


Fig. 6. Comparison of one-step ahead predictions for the L-ELM with OS-ELM and the linear models for the more complex system

C. Stability Advantage of L-ELM

It has been reported that ELM might run into an ill-posed problem when it is over parameterized or when not properly initialized [11], [12], [13], [14]. A few attempts have been made to improve the regularization behavior of ELM [13] and OS-ELM [14]. However, when the data is being processed 1-by-1 (as in the case of system identification), the regularization improvement suggested by [14] was not found to improve the situation. Such an unstable parametric evolution

can cause fatal problems when such online models are used in decision making. To address the issue, a Lyapunov based stable learning algorithm was developed for ELM models. The simulation of such a scenario is summarized in the Fig. 8 for the system considered in example II. It can be seen that with a small number of hidden neurons ($n_h=3$), the condition number (of matrix M in equation (11)) remains within reasonable values and the parameters do not diverge. However, as the number of hidden neurons increase, the condition number grows to a very high value creating an ill-conditioned least squares problem [11], the solution of which is absurd. For any general case, the solution of OS-ELM is never guaranteed to be stable. The Lyapunov based algorithm on the other hand, results in a model that is well regularized where parameter growth can be controlled using small values for the gain matrix Γ . The Lyapunov method gives a stability guarantee and performs well with no undesirable parameter growth even when the model is over-parameterized. Such a guarantee is necessary for control related applications. This shows the effectiveness of the method.

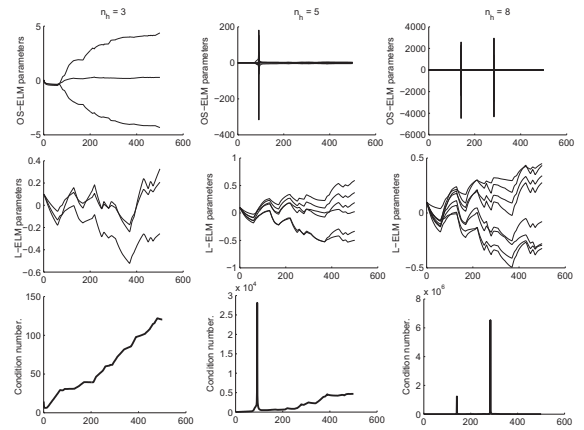


Fig. 8. The ill-conditioning of OS-ELM as more number of hidden neurons (n_h) are added compared to bounded parameter evolution of L-ELM.

V. CONCLUSIONS AND FUTURE WORK

The existing online learning ELM algorithm (OS-ELM) has an inherent ill-conditioning problem which results in poor regularization and more crucially instabilities in parameter evolution. A Lyapunov function was defined to develop a stable online learning algorithm for regression learning

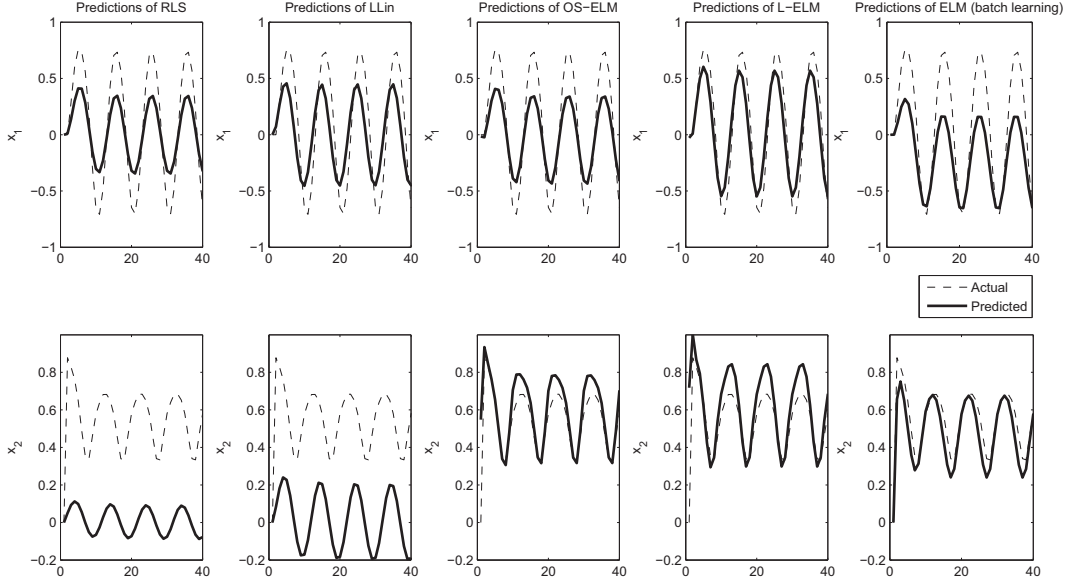


Fig. 7. Comparison of multi-step ahead predictions for the L-ELM with OS-ELM, O-ELM and the linear models for the more complex system

of systems connected in time (dynamic systems and time-series). Simulation results on simple examples demonstrate the issue with existing OS-ELM algorithm and validate the effectiveness of the developed algorithm. The comparison with an offline batch learning nonlinear ELM shows that the proposed online learning algorithm achieves good accuracy levels, better regularization and using much simpler computation suitable for real-time learning. Future work will focus on analysis of the algorithm in terms of local versus global convergence and application to real world systems.

ACKNOWLEDGMENT

This material¹ is based upon work supported by the Department of Energy and performed as a part of the ACCESS project consortium (Robert Bosch LLC, AVL Inc., Emitec Inc.) under the direction of PI Hakan Yilmaz, Robert Bosch, LLC.

APPENDIX: DERIVATION OF LYAPUNOV BASED PARAMETRIC UPDATE ALGORITHM

Consider a positive definite, decrescent and radially unbounded [8] function $V(e, \tilde{W}, k)$

$$V(e, \tilde{W}, k) = e^T(k)Pe(k) + \frac{1}{2}tr(\tilde{W}(k)\Gamma\tilde{W}^T(k))$$

¹Disclaimer: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

where P is a positive definite matrix $\in \mathbb{R}^{n \times n}$, n is the number of states of the system (13). Taking the difference in V ,

$$\begin{aligned} & V(k+1) - V(k) \\ &= e^T(k+1)Pe(k+1) + \frac{1}{2}tr(\tilde{W}(k+1)\Gamma\tilde{W}^T(k+1)) \\ &\quad - e^T(k)Pe(k) - \frac{1}{2}tr(\tilde{W}(k)\Gamma\tilde{W}^T(k)) \\ &= [Ae(k) + \tilde{W}^T(k)\phi(k) + \epsilon]^T Pe(k+1) \\ &\quad + \frac{1}{2}tr(\tilde{W}(k+1)\Gamma\tilde{W}^T(k+1)) \\ &\quad - e^T(k)Pe(k) - \frac{1}{2}tr(\tilde{W}(k)\Gamma\tilde{W}^T(k)) \\ &= e^T(k)A^T Pe(k+1) + \phi(k)^T \tilde{W}(k) Pe(k+1) + \epsilon^T Pe(k+1) \\ &\quad + \frac{1}{2}tr(\tilde{W}(k+1)\Gamma\tilde{W}^T(k+1)) \\ &\quad - e^T(k)Pe(k) - \frac{1}{2}tr(\tilde{W}(k)\Gamma\tilde{W}^T(k)) \\ &= e^T(k)A^T P(Ae(k) + \tilde{W}^T(k)\phi(k) + \epsilon) \\ &\quad + \phi(k)^T \tilde{W}(k) Pe(k+1) + \epsilon^T Pe(k+1) \\ &\quad + \frac{1}{2}tr(\tilde{W}(k+1)\Gamma\tilde{W}^T(k+1)) \\ &\quad - e^T(k)Pe(k) - \frac{1}{2}tr(\tilde{W}(k)\Gamma\tilde{W}^T(k)) \end{aligned}$$

$$\begin{aligned}
&= e^T(k)A^T PAe(k) + e^T(k)A^T P\tilde{W}^T(k)\phi(k) + e^T(k)A^T P\epsilon \\
&\quad + \phi(k)^T \tilde{W}(k)Pe(k+1) + \epsilon^T Pe(k+1) \\
&\quad + \frac{1}{2}tr(\tilde{W}(k+1)\Gamma\tilde{W}^T(k+1)) \\
&\quad - e^T(k)Pe(k) - \frac{1}{2}tr(\tilde{W}(k)\Gamma\tilde{W}^T(k))
\end{aligned}$$

Let $Q \in \mathbb{R}^{n \times n}$ be a positive definite matrix which satisfies the following discrete Lyapunov equation

$$A^T PA - P = -Q$$

$$\begin{aligned}
V(k+1) - V(k) &= -e^T(k)Qe(k) + e^T(k)A^T P\epsilon + \epsilon^T Pe(k+1) \\
&\quad + e^T(k)A^T P\tilde{W}^T(k)\phi(k) + \phi^T(k)\tilde{W}(k)Pe(k+1) \\
&\quad + \frac{1}{2}tr(\tilde{W}(k+1)\Gamma\tilde{W}^T(k+1)) - \frac{1}{2}tr(\tilde{W}(k)\Gamma\tilde{W}^T(k)) \\
&= -e^T(k)Qe(k) + e^T(k)A^T P\epsilon + \epsilon^T Pe(k+1) \\
&\quad + e^T(k)A^T P\tilde{W}^T(k)\phi(k) + \phi^T(k)\tilde{W}(k)Pe(k+1) \\
&\quad + \frac{1}{2}tr([\tilde{W}(k) + \Delta\tilde{W}(k)]\Gamma[\tilde{W}(k) + \Delta\tilde{W}(k)]^T) \\
&\quad - \frac{1}{2}tr(\tilde{W}(k)\Gamma\tilde{W}^T(k)) \\
&= -e^T(k)Qe(k) + e^T(k)A^T P\epsilon + \epsilon^T Pe(k+1) \\
&\quad + e^T(k)A^T P\tilde{W}^T(k)\phi(k) + \phi^T(k)\tilde{W}(k)Pe(k+1) \\
&\quad + tr(\Delta\tilde{W}^T(k)\Gamma\tilde{W}(k))
\end{aligned}$$

By converting, $e^T(k)A^T P\tilde{W}^T(k)\phi(k)$
 $= tr(PAe(k)\phi^T(k)\tilde{W}(k))$ and
 $\phi^T(k)\tilde{W}(k)Pe(k+1) = tr(Pe(k+1)\phi^T(k)\tilde{W}(k))$

$$\begin{aligned}
V(k+1) - V(k) &= -e^T(k)Qe(k) + e^T(k)A^T P\epsilon + \epsilon^T Pe(k+1) \\
&\quad + tr(\Delta\tilde{W}^T(k)\Gamma\tilde{W}(k)) \\
&\quad + PAe(k)\phi^T(k)\tilde{W}(k) + Pe(k+1)\phi^T(k)\tilde{W}(k)
\end{aligned}$$

By setting the terms in the trace to be zero, we get

$$\begin{aligned}
0 &= \Delta\tilde{W}^T(k)\Gamma\tilde{W}(k) \\
&\quad + PAe(k)\phi^T(k)\tilde{W}(k) + Pe(k+1)\phi^T(k)\tilde{W}(k)
\end{aligned}$$

$$\Delta\tilde{W}(k) = -\Gamma^{-T}\phi(k)[e(k+1) + Ae(k)]^T P$$

Also, $\tilde{W}(k) = W_* - \hat{W}(k)$, the change in parameter can be given by

$$\Delta\hat{W}(k) = \Gamma^{-T}\phi(k)[e(k+1) + Ae(k)]^T P$$

$$\begin{aligned}
\hat{W}(k+1) &= \hat{W}(k) + \Delta\hat{W}(k) \\
&= \hat{W}(k) + \Gamma^{-T}\phi(k)[e(k+1) + Ae(k)]^T P
\end{aligned}$$

REFERENCES

- [1] N. Liang, G. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *Neural Networks, IEEE Transactions on*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [2] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.
- [3] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, ser. Engineering Online Library. Springer, 2001.
- [4] V. M. Janakiraman, X. Nguyen, and D. Assanis, "Nonlinear identification of a gasoline HCCI engine using neural networks coupled with principal component analysis," *Applied Soft Computing*, vol. 13, no. 5, pp. 2375 – 2389, 2013.
- [5] —, "A system identification framework for modeling complex combustion dynamics using support vector machines," in *Informatics in Control, Automation and Robotics*, ser. Lecture Notes in Electrical Engineering. Springer Berlin / Heidelberg, 2013.
- [6] K. S. Narendra and S. Mukhopadhyay, "Adaptive control using neural networks and approximate models," in *IEEE Transactions on Neural Networks*, ser. 3, vol. 8, May 1997.
- [7] V. Akpan and G. Hassapis, "Adaptive predictive control using recurrent neural network identification," in *Control and Automation, 2009. MED '09. 17th Mediterranean Conference on*, june 2009, pp. 61 – 66.
- [8] P. Ioannou and J. Sun, *Robust adaptive control*.
- [9] L. Yan, N. Sundararajan, and P. Saratchandran, "Nonlinear system identification using lyapunov based fully tuned dynamic rbf networks," *Neural Process. Lett.*, vol. 12, no. 3, pp. 291–303, Dec. 2000.
- [10] P. Singla and J. Junkins, *Multi-Resolution Methods for Modeling and Control of Dynamical Systems*, ser. Chapman & Hall/CRC Applied Mathematics & Nonlinear Science. Taylor & Francis, 2010.
- [11] G. Zhao, Z. Shen, C. Miao, and Z. Man, "On improving the conditioning of extreme learning machine: A linear case," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, dec. 2009, pp. 1 – 5.
- [12] F. Han, H.-F. Yao, and Q.-H. Ling, "An improved extreme learning machine based on particle swarm optimization," in *Bio-Inspired Computing and Applications*, ser. Lecture Notes in Computer Science.
- [13] M. T. Hoang, H. Huynh, N. Vo, and Y. Won, "A robust online sequential extreme learning machine," in *Advances in Neural Networks*, ser. Lecture Notes in Computer Science.
- [14] H. T. Huynh and Y. Won, "Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks," *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1930 – 1935, 2011.
- [15] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2006.
- [16] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 42, no. 2, pp. 513–529, 2012.
- [17] S. Lyashevskiy and L. Abel, "Nonlinear systems identification using the lyapunov method," *System identification (SYSID'94) : a postprint volume from the IFAC symposium, Copenhagen, Denmark, 4-6 July 1994*, vol. 1, July 1994.
- [18] K. Ross, *Elementary Analysis: The Theory of Calculus*, ser. Undergraduate Texts in Mathematics. Springer, 1980.
- [19] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.
- [20] R. Nowak and B. Van Veen, "Nonlinear system identification with pseudorandom multilevel excitation sequences," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, vol. 4, april 1993, pp. 456 – 459 vol.4.