Final Report

for

Identification and Control of Aircrafts using Multiple Models and Adaptive Critics

NASA Proposal No.: 04-0617

Coordinator: Dr. Mark Motter

Period of Performance
4/27/04 – 10/26/06

PI: Jose C. Principe

University of Florida

April 2007

Proposed work

This was a continuation proposal to test and implement a novel multiple model framework for nonlinear plants whose dynamics are rapidly varying in a state space that is assumed known. During this research we had the opportunity to compare in the same dataset two parallel approaches to implement the gating function in the multiple model framework. We also had the opportunity to evaluate a new and intriguing nonlinear dynamical system approach called the echo state network (ESN) to implement adaptive critics. Unfortunately, we had to stop at the system identification stage, because a no cost extension request was not granted.

Productivity

During the time of this grant we published three journal papers and one conference proceedings, acknowledging the NASA support:

Cho J., Principe P., Erdogmus D. and Motter M., "Quasi-Sliding Mode Control Strategy Based on Multiple-Linear Models," NeuroComputing, Volume 70, Issues 4-6, January 2007, Pages 960-974

Cho J., Principe P., Erdogmus D. and Motter M., "Modeling and Inverse Controller Design for an Unmanned Aerial Vehicle Based on the Self-Organizing Map," IEEE Transactions on Neural Networks, Volume 17, Issue 2, March 2006 Page(s):445 - 460.

Lan J., Cho J., Erdogmus D., Principe J., Motter M., Xu J., "Local Linear PID Controllers for Nonlinear Control", Int. J. of Control and Intelligent. Systems, Special Issue Nonlinear Adaptive PID Controls, vol 33, #1, 26-35, 2005

Xu D., Lan J. Principe J., “Direct Adaptive Control: An Echo State Network and Genetic Algorithm Approach” IEEE Int. Joint Conf. on Neural Networks (IJCNN05), Montreal, Canada.

Summary of the findings

We compared two possible implementations of local linear models for control: one approach is based on a self-organizing map (SOM) to cluster the dynamics followed by a set of linear models operating at each cluster. Therefore the gating function is hard (a single local model will represent the regional dynamics). This simplifies the controller design since there is a one to one mapping between controllers and local models. The second approach uses a soft gate using a probabilistic framework based on a Gaussian Mixture Model (also called a dynamic mixture of experts). In this approach several

models may be active at a given time, we can expect a smaller number of models, but the controller design is more involved, with potentially better noise rejection characteristics.

Our experiments showed that the SOM provides overall best performance in high SNRs, but the performance degrades faster than with the GMM for the same noise conditions. The SOM approach required about an order of magnitude more models than the GMM, so in terms of implementation cost, the GMM is preferable. The design of the SOM is straight forward, while the design of the GMM controllers, although still reasonable, is more involved and needs more care in the selection of the parameters. Either one of these locally linear approaches outperform global nonlinear controllers based on neural networks, such as the time delay neural network (TDNN). Therefore, in essence the local model approach warrants practical implementations. In order to call the attention of the control community for this design methodology we extended successfully the multiple model approach to PID controllers (still today the most widely used control scheme in the industry), and wrote a paper on this subject.

The echo state network (ESN) is a recurrent neural network with the special characteristics that only the output parameters are trained. The recurrent connections are preset according to the problem domain and are fixed. In a nutshell, the states of the "reservoir" of recurrent processing elements implement a projection space, where the desired response is optimally projected. This architecture trades training efficiency by a large increase in the dimension of the recurrent layer. However, the power of the recurrent neural networks can be brought to bear on practical difficult problems. Our goal was to implement an adaptive critic architecture implementing Bellman's approach to optimal control. However, we could only characterize the ESN performance as a critic in value function evaluation, which is just one of the pieces of the overall adaptive critic controller. The results were very convincing, and the simplicity of the implementation was unparalleled.

Result Highlights

SOM-Based Competitive LLM

Suppose that input-output data pairs of the form $\{(u_1,y_1),\ldots,(u_N,y_N)\}$, where u is the input signal and y is the output signal, is available from a SISO system for system identification. It is assumed that the system that generated this data sequence is of the following general nonlinear time-invariant form (assuming that there is no instantaneous effect of the input on the output):

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1} &= f(\tilde{\mathbf{x}}_k, u_k) \\ y_k &= h(\tilde{\mathbf{x}}_k) \end{aligned} \qquad (1)$$

In the state space, this nonlinear dynamical equation can be approximated by an ordinary linear differential equation of the form:

$$y_k = a_1 y_{k-1} + \ldots + a_n y_{k-n} + b_1 u_{k-1} + \ldots + b_m u_{k-m} \qquad (2)$$

The coefficient vectors **a** and **b** depend on the local regime; consequently system identification using multiple local linear models of this form necessitate the quantization

of the state space. The quantization can be adaptively achieved by training a SOM on the combined state vector $\mathbf{x}$ defined as

$$\mathbf{x}_k = [y_{k-1} \quad \cdots \quad y_{k-n} \quad u_{k-1} \quad \cdots \quad u_{k-m}]^T \quad (3)$$

The SOM consists of a multidimensional array of neurons with weight vectors $\mathbf{w}_i$, which are trained competitively on the input vector samples $\mathbf{x}_k$. While the samples are presented to the SOM one at a time, in a series of epochs, the weight vector of the winner neuron $\mathbf{w}_w$ for the particular sample (i.e. the weight vector that minimizes the Mahalanobis distance $d(\mathbf{w}, \mathbf{x}_k) = (\mathbf{w} - \mathbf{x}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{w} - \mathbf{x}_k)$ between the two vectors) and its topological neighbors $\mathbf{w}_n$ are updated according to the following rules [13].

$$\begin{aligned} \mathbf{w}_w(t+1) &\leftarrow \mathbf{w}_w(t) + \eta(t)\boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \mathbf{w}_w(t)) \\ \mathbf{w}_n(t+1) &\leftarrow \mathbf{w}_n(t) + \eta(t) \cdot \\ & \quad h(\| \mathbf{w}_n(t) - \mathbf{w}_w(t) \|, \sigma(t))\boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \mathbf{w}_n(t)) \end{aligned} \quad (4)$$

In (4), the (Gaussian) neighborhood function $h(.,\sigma)$ is annealed in time such that initially the neighborhood radius σ covers most of the network and towards the end of training it is narrow enough to include only the winner neuron. Typically linear or exponential annealing schemes are used for both the neighborhood radius and the step size η.

Once the SOM is trained, the available data is partitioned into smaller sets; the samples associated with weight vector $\mathbf{w}_i$ are $\{(\mathbf{x}_{i1}, y_{i1}),...,(\mathbf{x}_{iN_i}, y_{iN_i})\}$, where N_i is the number of samples clustered to neuron i. Each neuron also has a linear models attached to each it, whose coefficient vectors $\mathbf{a}_i$ and $\mathbf{b}_i$ can be optimized using least squares with this input-output training data that is clustered to this neuron in the SOM. The overall LLM output can be produced either using hard competition or soft competition between the linear models associated with each neuron. In particular, the model output is determined as follows:

$$\hat{y}_k = \sum_{i=1}^{M} p_{ik} [\mathbf{a}_i^T \quad \mathbf{b}_i^T] \mathbf{x}_k \quad (5)$$

For hard competition, the scaling factor p_{ik} is either 1 or 0 depending on the i^{th} neuron being the winner or not for input $\mathbf{x}_k$. For soft competition, which also allows a smoother transition between the linear models at switching boundaries, the model output can be determined by a weighted average. In this case, one can use, for example, a weighting based on the Mahalanobis distance again:

$$p_{ik} = d(\mathbf{w}_i, \mathbf{x}_k) \Big/ \sum_{j=1}^{M} d(\mathbf{w}_j, \mathbf{x}_k) \quad (6)$$

Alternatively, similar weighting schemes where only a few of the nearest models are activated could be employed. From a probabilistic point-of-view, such a weighting mechanism can be regarded as the probability of the corresponding linear model being responsible for the sample under consideration. The GMM approach described in the following section makes explicit use of this probabilistic view for the data.

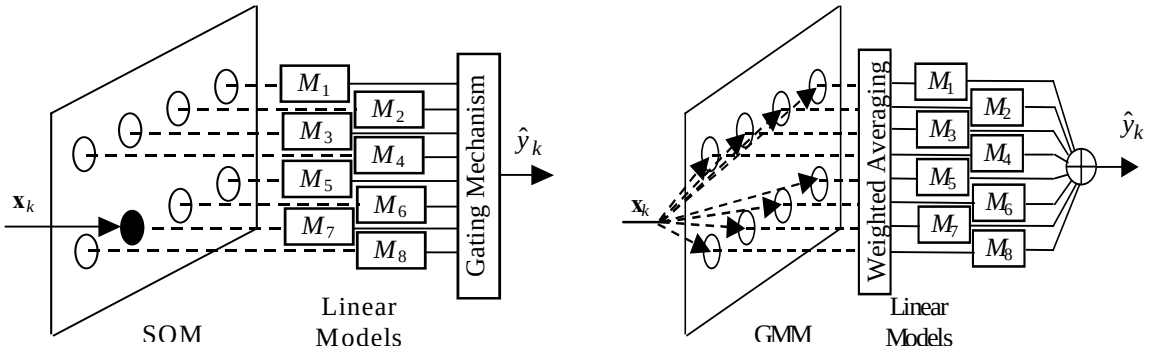


Figure 1a. (left panel) Schematic diagram of the SOM-based LLM approach for system identification. The winner neuron is illustrated by the full circle.
Figure 1b. (right panel) Schematic diagram of the GMM-based LLM approach for system identification.

GMM-Based Cooperative LLM

Suppose that input-output data pairs of the form $\{(u_1,y_1),\ldots,(u_N,y_N)\}$ are available from a SISO system as described in (1). The linear approximation in (2) still holds locally. The representative state vector is again described as in (3). In the GMM approach, instead of partitioning this vector into disjoint sets using a SOM, it is assumed that the data is generated by a multi-dimensional joint mixture of Gaussian distribution. In other words, the distribution of vector $\mathbf{x}$ is given by

$$p(\mathbf{x}_k) = \sum_{i=1}^{M} \alpha_i G(\mathbf{x}_k; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \qquad (7)$$

where $G(\mathbf{x};\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian density with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The coefficient α_i denotes the probability of occurrence of this mode in the GMM. Given the training data, once the vector samples $\mathbf{x}_k$ are constructed, the maximum likelihood solution for the parameters α_i, $\boldsymbol{\mu}_i$, and $\boldsymbol{\Sigma}_i$ can be determined using the expectation maximization (EM) algorithm [14]. Alternative statistical model selection approaches could also be employed. As in the SOM-based model, here it is assumed that a linear model is associated with each of the Gaussian model in the GMM. Each linear model has weight vectors $\mathbf{a}_i$ and $\mathbf{b}_i$ that can be optimized using least squares once the parameters of the GMM are determined using the EM algorithm. The overall model output is a weighted average of individual linear models

$$\hat{y}_k = \sum_{i=1}^{M} p_{ik} [\mathbf{a}_i^T \quad \mathbf{b}_i^T] \mathbf{x}_k = \boldsymbol{\theta}^T \mathbf{z}_k \qquad (8)$$

where $p_{ik} = \alpha_i G(\mathbf{x}_k; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. The overall linear model weight vector is defined as $\boldsymbol{\theta} = [\mathbf{a}_1^T \quad \mathbf{b}_1^T \quad \cdots \quad \mathbf{a}_M^T \quad \mathbf{b}_M^T]$, and the corresponding overall modified input vector is $\mathbf{z}_k = [p_{1k}\mathbf{x}_k^T \quad \cdots \quad p_{Mk}\mathbf{x}_k^T]$. The linear models can be optimized simultaneously using least squares considering the available output measurements y_k as desired outputs corresponding to the modified input sample $\mathbf{z}_k$. Fig 1 compares the block diagrams fo3 the two implementations for a graphical display of the differences.

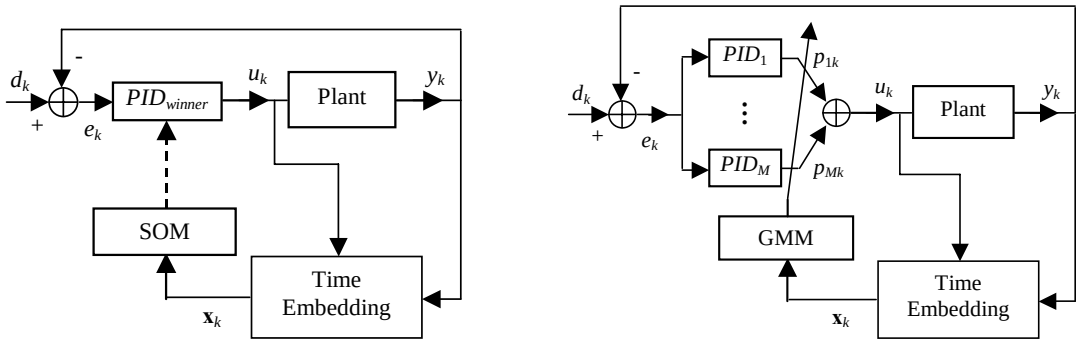


Figure 2a. Schematic diagram of the switching linear PID controllers based on the SOM decision for the winner model.
Figure 2b. Schematic diagram of the weighted average PID controller based on the GMM assessment of contributions.

Local Linear PID Controller Design

To illustrate how to develop controllers for these architectures, the PIDs will be utilized since they are so widely used. Notice that here a globally nonlinear control action can be implemented, from the set of local linear models. The literature has an abundance of PID design methodologies for linear SISO systems including direct pole-placement techniques and optimal coefficient adjustment according to some criteria [24,25]. In the case of MIMO systems, where the system identification phase is carried out as described in the previous sections with the necessary extensions and modifications to the SOM and GMM approach, one can resort to MIMO PID design techniques [26].

Given the local linear models as obtained through the use of a SOM or a GMM, and a PID design technique, the overall closed loop nonlinear PID design reduces to determining the coefficients of the individual local linear PID controllers using their respective linear plant model transfer functions. For both SOM-based LLM and GMM-based LLM methodologies, one needs to determine a set of PID coefficients per linear model. In the competitive SOM approach, the model output depends only on a single linear model at a given time; therefore, the PID coefficients are set to those values determined for the instantaneous winner model (as illustrated in Fig. 2a). In the GMM approach, since the models are averaged with probabilistic weights, the same must be done with the PID controllers. An alternative look at (8) shows that the GMM-based LLM output is also equivalently expressed as

$$\hat{y}_k = \mathbf{x}_k^T \left(\sum_{i=1}^{M} p_{ik} \begin{bmatrix} \mathbf{a}_i \\ \mathbf{b}_i \end{bmatrix} \right) = \mathbf{x}_k^T \begin{bmatrix} \bar{\mathbf{a}}_{ik} \\ \bar{\mathbf{b}}_{ik} \end{bmatrix} \qquad (9)$$

where $\bar{\mathbf{a}}_{ik}$ and $\bar{\mathbf{b}}_{ik}$ denote the weighted average linear model at time instant k. Consecutively, The instantaneous PID controller will be characterized by a weighted average of the PID coefficients of all linear models as follows:

$$PID_k = \sum_{i=1}^{M} p_{ik} PID_i \qquad (10)$$

where PID_i is the transfer function of the PID controller for the i^{th} local model (as illustrated in Fig. 2b). In the case of a soft combination SOM-based LLM as described by (6), the formulation in (10) can be employed for the PID controller.

Simulations

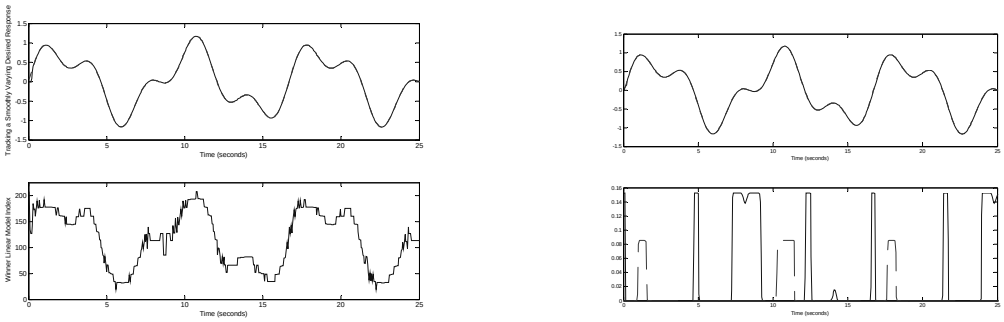


Figure 3a. Response of the closed loop nonlinear PID control system to smoothly varying desired response is shown at the top for the SOM-based LLM approach. The bottom plot shows the index of the winner linear model changing in time as determined by the SOM based on the instantaneous input.
Figure 3b. Response of the closed loop nonlinear PID control system to various step changes in the desired response is shown at the top for the GMM-based LLM approach. The bottom plot shows the contribution coefficients of the linear models changing in time as determined by the Gaussian mixture model based on the instantaneous input.

The performance of the proposed local linear PID design approach will be demonstrated on a simplified SISO nonlinear missile dynamic model where only the yaw dynamics are considered [5]. The simplified two-state dynamics are described by:

$$\begin{aligned} \dot{x}_1 &= x_2 - 0.1\cos(x_1)(5x_1 - 4x_1^3 + x_1^5) - 0.5\cos(x_1)u \\ \dot{x}_2 &= -65x_1 + 50x_1^3 - 15x_1^5 - x_2 100u \\ y &= x_1 \end{aligned} \qquad (11)$$

Since the input is the rudder deflection, it is limited by ±0.5 radians in practice. The simulations invoke this limit on the control input value although the PID design will not take this into consideration.

Two models were identified using 6000 samples of input-output pairs obtained from the dynamical equation (11) using a discretization time step of T_s=0.05s (which corresponds to 300s of flight time). In order to excite a rich variety of dynamical modes in the plant, the system identification data was generated using a uniformly random input signal within the specified limits. The embedding delays for the input and the output were both selected to be 2, resulting in 4-coefficient linear models. This embedding delay was intuitively chosen in accordance with the dimensionality of the state dynamics, but was also tested experimentally. Increasing the embedding delay did not result in system identification performance improvement. The SOM-based LLM approach utilized a 15x15 rectangular grid of neurons, amounting to 225 competitive local linear models, whereas the GMM-based LLM approach assumed to 5-mode mixture model, resulting in 5 cooperative linear models. The identified models were also tested on original 50s-length data (1000 samples), generated using a new sequence of random input. The actual plant output, the model predicted output and the estimation error for the SOM-based LLM and GMM-based LLM models are provided in Fig. 3a and Fig. 3b. The signal-to-error ratio (SER) for these two models on the testing set are found to be 31.7dB and 31dB, respectively.

The PID controller design is carried out using the standard pole-placement technique [24]. For both modeling methodologies and all linear models, the corresponding PID coefficients are determined to bring the closed-loop response poles from the plant output

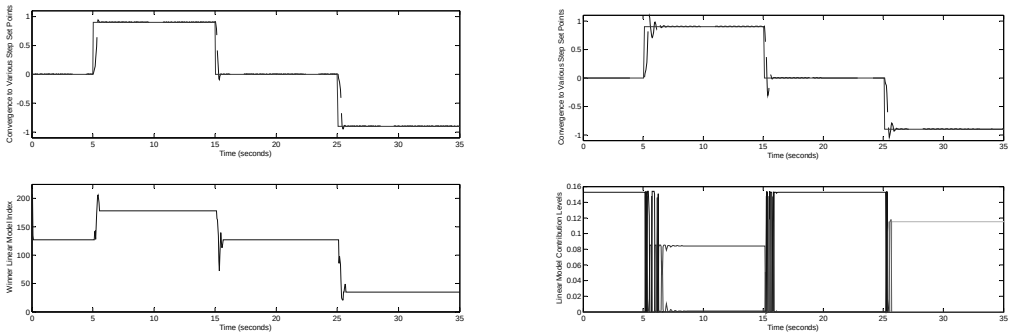


Figure 4a. Response of the closed loop nonlinear PID control system to various step changes in the desired response is shown at the top for the SOM-based LLM approach. The bottom plot shows the index of the winner linear model changing in time as determined by the SOM based on the instantaneous input.
Figure 4b. Response of the closed loop nonlinear PID control system to various step changes in the desired response is shown at the top for the GMM-based LLM approach. The bottom plot shows the contribution coefficients of the linear models changing in time as determined by the Gaussian mixture model based on the instantaneous input.

to the desired output to 0, 0, 0.05+*i*0.3, and 0.05-*i*0.3. This closed-loop nonlinear PID controller is tested in two cases: one with various step changes to the desired output (Fig. 4a and Fig. 4b).

The step and tracking performance of the designed nonlinear PID control scheme in closed-loop operation with the actual plant model is very satisfactory when compared with the performance of an adaptive nonlinear time-delay neural network (TDNN, not shown). However, there are differences in terms of the switching, where the GMM is achieving the same control law by switching very fast (chattering) among the models during the transitions, while the SOM achieves a smooth transition. We hope this example illustrates well the plus and minus of each control architecture with controllers that are well known in the community.

System ID based on ESN model.

Echo State Networks (ESN) first proposed by Jaeger are a new kind of recurrent networks with simplified learning mechanisms which largely decrease training time. The state model of an ESN can be written as

$$\begin{cases} S_{k+1} = f(W_{in} \cdot X + W \cdot S_k); \\ Y_{k+1} = W_{out} \cdot S_{k+1} \end{cases} \tag{12}$$

where W_{in} is input weights, W is internal weights, W_{out} is output weights, S represent the internal states, X and Y are corresponding to system input and output. The ESN has a large "reservoir" recurrent network that can produce diversified representations of an input signal, which can then be instantaneously combined into an optimal manner to approximate a desired response. As a competitor of time delay embedding methods (like the delay line used in both the SOM and GMM), ESN captures the dynamics in the recurrent connections. The input vector is connected to a "reservoir" of N discrete-time recurrent networks by a connection matrix W_{in}. At any time instant k, the readout S_k (state output) from the RNN reservoir is a column vector. The reservoir states are transformed

by a static linear mapper. Each processing element (PE) in the reservoir can be implemented as a leaky integrator and the state output or the readout is given by the difference equation in (12).

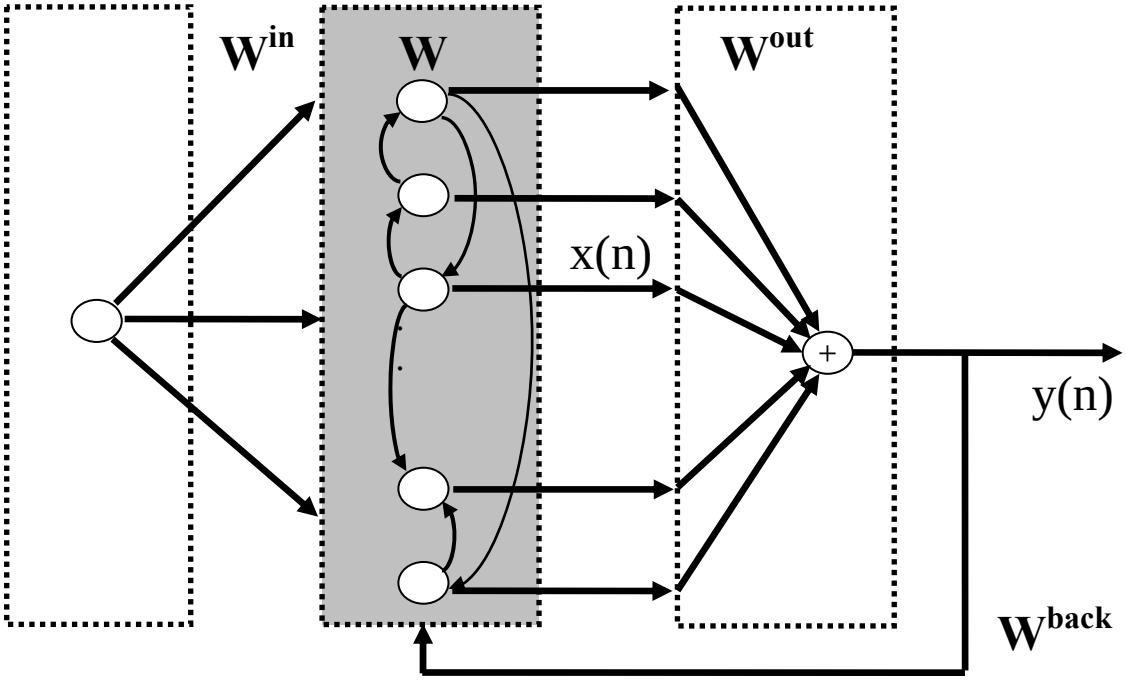


Fig.5, Block diagram of ESN

System ID through ESN

The big advantage of the ESN is that only the output weights are trained, which represent a training problem linear in the weights, for which even analytic techniques (least squares) are available. Therefore, the training becomes much easier and also much fewer samples are required to accomplish it.

Figure 6 compares the system identification performance on the Lorenz system and the missile system, and illustrates well that the ESN is capable of making a good state representation even without using time embedding. The prediction performance is better for the ESN than that from time-embedding data (on an average of around 3dB). For the LoFlyte system, since the performance from LLM has been over 40 dB, there is no 3dB advantage from ESN (SER from ESN is 40.8379 dB), yet the input dimension for ESN is decreased from eight (one step embedding) or 12 (two steps embedding) to four (only current state). The training time is dramatically decreased. The promising aspect of the dimension decreasing and fast training is not only reflected on the simplification of algorithm, it will save time and resource for on-line control as well.

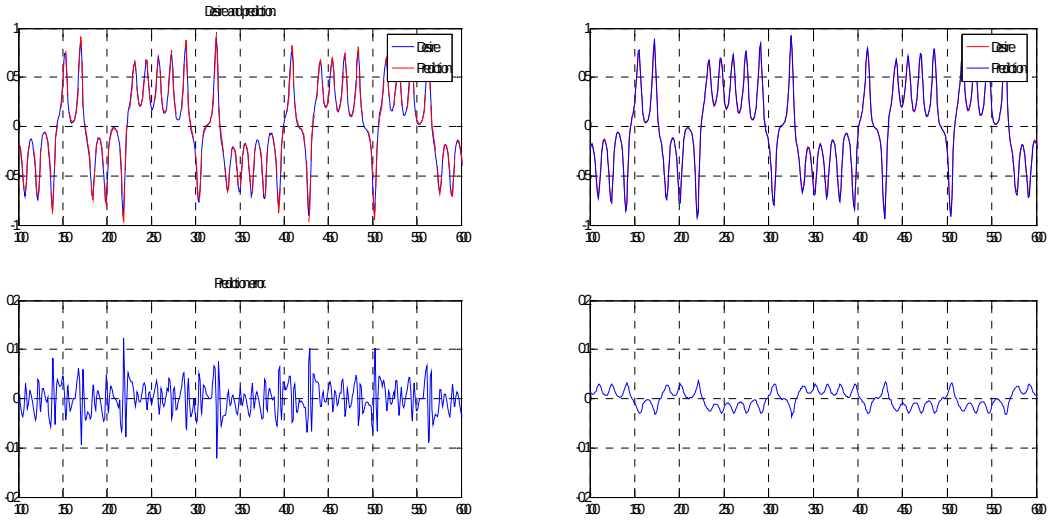
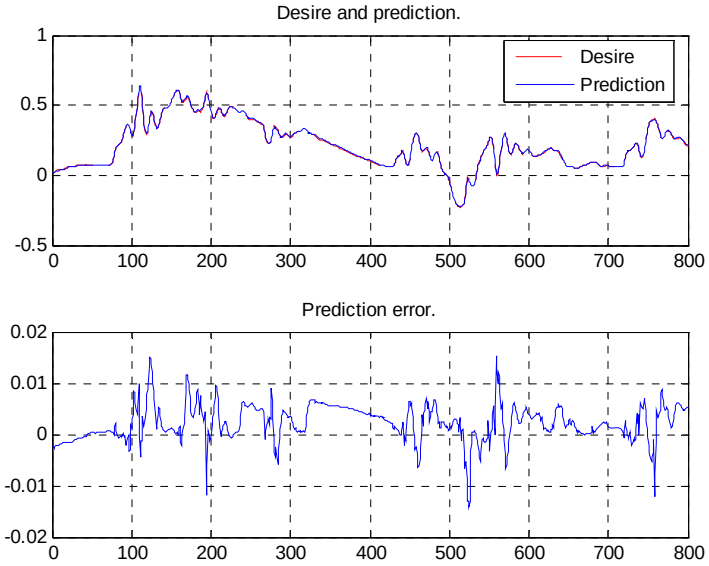
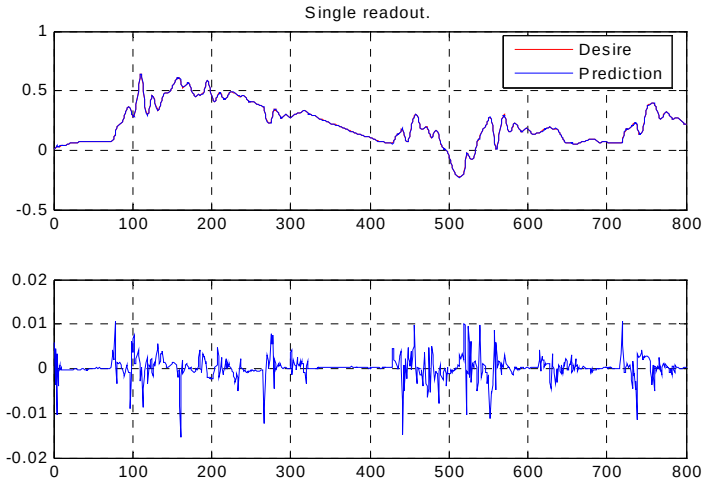


Fig.6, Left: Prediction performance from GMM-LLM (two-step embedding); Right: Prediction performance from ESN (zero embedding).



(a)



(b)

Fig. 7, (a): Prediction performance from GMM-LLM for the LoFlyte with embedding; (b): Prediction performance from ESN for the LoFlyte without embedding.

REFERENCES

[1] O. Nelles, *Nonlinear System Identification*, Springer, New York, 2001.

[2] I.J. Leontaritis, S.A. Billings, "Input-Output Parametric Models for Nonlinear Systems Part I: Deterministic Nonlinear Systems," International Journal of Control, vol. 41, no. 2, pp. 303-328, 1985.

[3] K.S. Narendra, K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," IEEE Trans. Neural Networks, vol. 1, no. 1, pp. 4-27, 1990.

[4] T.A. Johansen, B.A. Foss, "Constructing NARMAX Models Using ARMAX Models," International Journal of Control, vol. 58, no. 5, pp. 1125-1153, 1993.

[5] X. Ni, M. Verhaegen, A.J. Krijgsman, H.B. Verbruggen, "A New Method for Identification and Control of Nonlinear Dynamic Systems," Engineering Applications of Artificial Intelligence, vol. 9, no. 3, pp. 231-243, 1996.

[6] D.M. Walker, N.B. Tufillaro, P. Gross, "Radial-Basis Models for Feedback Systems with Fading Memory," IEEE Transactions on Circuits and Systems, vol. 48, no. 9, pp. 1147-1151, 2001.

[7] B.S. Kim, A.J. Calise, "Nonlinear Flight Control Using Neural Networks," Journal of Guidance, Control, and Dynamics, vol. 20, no. 1, pp. 26-33, 1997.

[8] C. H. Lee and M. J. Chung, "Gain-Scheduled State Feedback Control Design Technique for Flight Vehicles," IEEE Transactions on Aerospace and Electronic Systems, vol. 37, no. 1, pp. 173-182, 2001.

[9] R.C. Dorf, R.H. Bishop, *Modern Control Systems*, 8th ed., Addison, Wesley, New York, 1998.

[10] J.S.R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 23, no. 3, pp. 665-685, 1993.

[11] J. Lan, J.C. Principe, M.A. Motter, "Identification of Dynamical Systems Using GMM with VQ Initialization," Proceedings of IJCNN'03, vol. 1, pp. 764-768, 2003.

[12] J.C. Principe, L. Wang and M.A. Motter, "Local Dynamic Modeling with Self-Organizing Maps and Applications to Nonlinear System Identification and Control," Proceedings of IEEE, vol. 86, no. 11, pp. 2240-2258, 1998.

[13] T. Kohonen, *Self-Organizing Maps*, Springer, New York 1995.

[14] G.J.Mclachlan, D. Peel, *Finite Mixture Models*, Wiley, New York, 2001.

[15] J. Stark, D.S. Broomhead, M.E. Davies, J. Huke, "Takens Embedding Theorems for Forced and Stochastic Systems," Nonlinear Analysis, Theory Methods, and Applications, vol. 30, no. 8, pp. 5303-5314, 1997.

[16] M. Casdagli, “Nonlinear Prediction of Chaotic Time Series,” Physica. D, vol. 35, no. 3, pp. 35-356, 1989.
[17] F. Takens, “On Numerical Determination of the Dimension of an Attractor,” in *Dynamical Systems and Turbulance*, (D. Rand, L.S. Young eds.), Warwick 1980, *Lecture Notes in Mathematics*, vol. 898, pp. 366-381, Springer-Verlag, Berlin, 1981.
[18] K.S. Narendra and J. Balakrishnan, “Adaptive Control Using Multiple Models,” IEEE Transactions on Automatic Control, vol. 42, no. 2, pp. 171-187, 1997.
[19] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, “Adaptive Mixtures of Local Experts,” Neural Computation, vol. 3, pp. 79-87, 1991.
[20] R. Murray-Smith, T.A. Johansen, *Multiple Model Approaches to Modeling and Control*, Taylor & Francis, New York, 1997.
[21] M.A. Motter, *Control of the NASA Langley 16-foot Transonic Tunne; with the Self-Organizing Feature Map*, Ph.D. Dissertation, University of Florida, Gainesville, Florida, 1997.
[22] G. Thampi, J.C. Principe, M.A. Motter, J. Cho, J. Lan, “Multiple Model Based Flight Control Design,” Proceedings of MWSCAS’02, vol.3, pp. 133-136, 2002.
[23] J. Cho, J. Lan, G. Thampi, J.C. Principe, M.A. Motter, “Identification of Aircraft Dynamics Using a SOM and Local Linear Models,” Proceedings of MWSCAS’02, vol. 2, pp. 148-151, 2002.
[24] R.E. Brown, G.N. Maliotis, J.A. Gibby, “PID Self-Tuning Controller for Aluminum Rolling Mill,” IEEE Transactions on Industry Applications, vol. 29, no. 3, pp. 578-583, 1993.
[25] K.M. Vu, “Optimal Setting for Discrete PID Controllers,” IEE Proceedings-D, vol. 139, no. 1, pp. 31-40, 1992.
[26] J. Bao, J.F. Forbes, P.J. McLellan, “Robust Multiloop PID Controller Design: A Successive Semidefinite Programming Approach,” Industrial and Engineering Chemistry Research, vol. 38, pp. 3407-3419, 1999.
[27] S. Haykin, Neural Networks A Comprehensive Foundation, Prentice Hall, 1998.