Adaptive Activity Driven Multi-Level Hierarchical Prediction of Complex Systems Through Profiling and Feedback

O.Hammami
ENSTA ParisTech
32 Bvd Victor
75739 Paris Cedex
FRANCE
hammami@ensta.fr

Abstract. Complex systems composed of multi-level hierarchical systems exhibit complex interacting dependences which affects the performance of the overall system. Although, one solution is to tightly monitor each system activity and feedback to the next step level in order to take appropriate measures and actions to improve the system this solution is costly and too systematic. In addition, a dependency chain exists between low level systems and higher level systems this dependency chains being as long as the number of levels of the system. We propose in this paper an activity driven adaptive hierarchical prediction technique for complex systems which minimizes monitoring and prediction resources requirements and still keep efficient overall systems performance prediction.

Introduction

Processes such as IEEE-Std-1220 and MIL-Std-499 are very important processes for systems analysis and control [Sage 2007]. However, they are based on the assumption of relatively stable systems with well identified and analyzed requirements. On the contrary, complex systems may exhibit changing behavior with varying functionnalities. In this context, continuous profiling, monitoring is necessary to adjust to the new system requirements. In his regard, system behavior prediction is paramount to anticipate futur requirements. Complex systems composed of multi-level hierarchical systems exhibit complex interacting dependences which affects the performance of the overall system. Although, one solution is to tightly monitor each system activity and feedback to the next step level in order to take appropriate measures and actions to improve the system this solution is costly and too systematic. In addition, a dependency chain exists between low level systems and higher level systems this dependency chains being as long as the number of levels of the system. This dependency chain has its own delay which might affect the capacity of the systems to react in appropriate time framework. We propose in this paper an activity driven adaptive prediction technique for complex systems which minimizes monitoring and prediction resources requirements and still keep efficient overall systems performance prediction.

Hierarchical Complex Systems

Hierarchical complex systems are composed of a collection of systems interconnected hierarchically. Each system has its own inputs and outputs with some outputs being inputs to the immediate upper level .The number of systems connected to an upper systems is not limited although obviously finite and the general structure needs not to be symmetric in the levels or hierarchy depth.

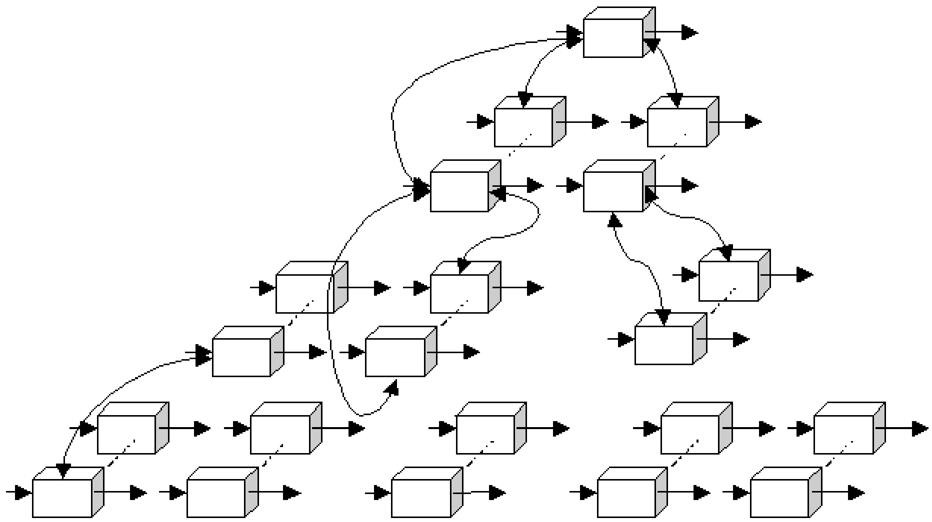

Figure 1. Hierarchical Complex systems

For a complex system such as the example above it becomes quickly difficult to monitor and control when the hierarchy level increases. In addition, some part of the system may have a more important varying activity requiring a tighter monitoring and control. The following figure gives an example of a small hierarchical system composed of 5 systems to which we add a monitoring and prediction unit. We assume in this example that the shaded box represents a system with very little varying activity while other systems have a more varying activity which requires tight monitoring and prediction in order to adapt the system.

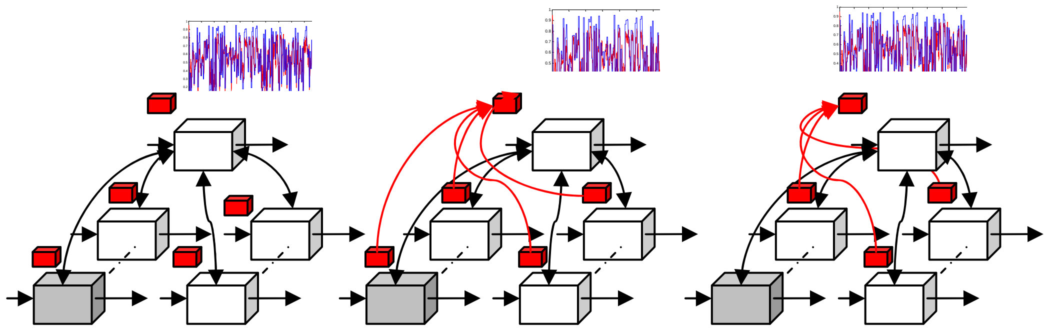

Figure 2. Monitor and prediction (1) exhaustive (2) hierarchical prediction (3) adaptive hierarchical prediction

In the first approach monitoring and prediction units are unconnected and it is assumed that the global monitoring and prediction is realized by some ad hoc technique. In the second case the global monitoring and prediction unit is hierarchically fed by the the lower level monitoring and prediction units even if some units do not contribute to the variations due to some very stable activity. The third approach is an adaptive technique which we only monitor systems which exhibit varying behavior and require prediction in order to contribute to the overall system behavior prediction. Several techniques have been used for prediction and forecasting of time series coming from applied mathematics and statistics. However, neural networks have been successfully used for the same purpose and have competed favorably with traditionnal statistical techniques. They have been used for various purposes. For examples, Barr [Barr 2008] have used them for organization, learning and cooperation, Bode [Bode 1998] have used them for decision support in the management of research and development, Breitler [Breitler 2004] for predicting reliability of complex systems, On Cheung [On Cheung 2006] have used them for predicting project performance, Selby [Selby 2006] for prediction in large scale systems engineering while William Chien [Chien 1999] have used them for strategic planning. Finally, they have been intensively used in embedded systems (example [Chtourou 2006]).

Neural Network Prediction

A recurrent neural network corresponds to a dynamic system composed by many states that evolve according to a number of nonlinear equations. Recurrent neural networks have been intensively used for dynamic process modeling [Nerrand 1994] and time series prediction [Lin 1996][Connor 1994]. The overall design process of a neural network is given in the below figure.

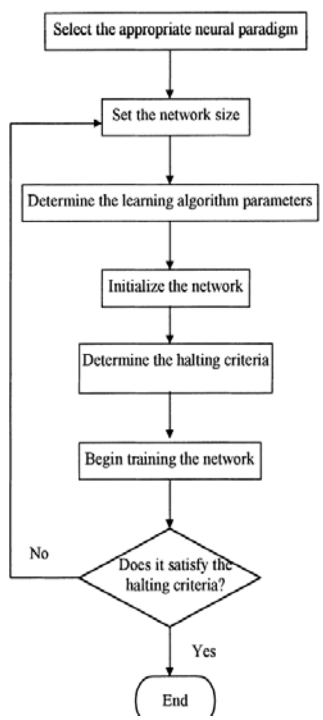

Figure 3. Neural network construction

A large variety of learning algorithms for recurrent neural networks have been explored and multiple architectures have been employed to solve the one step-ahead prediction problem such as the incremental Simple Recurrent Network (SRN) and Simple Recurrent Network with Shortcut Connections (SRNSC) implemented in to predict samples of the training sequences. In the SRN architecture, the recurrent inputs are connected to the neurons of the hidden layer. The SRNSC architecture is a SRN model with connections between neurons of input and output layers. Other RNN architectures have been proposed for implementing multi step ahead prediction system, which are the NARX model [Lin 96] and a dynamic recurrent neural network. The connections in the dynamic recurrent networks are composed by feedforward links, recurrent links (connection between neurons of the hidden layer) and cross-talk links (connection between neurons of the output and the hidden layers) .

NARX Neural Network architecture

Recurrent neural networks are able to model complex behaviors of dynamic systems and fully connected recurrent neural network have been proven to be computationally rich [Atiya 2000].

The mathematical formulation of the single-step-ahead prediction performed on the y time series values which in our case represents subsystem activity is given by the following equation:

$$\hat{y}(k+1) = p[y(k), \quad y(k-1), ..., y(k-N)] \qquad (1)$$

N is the length of the input vector of the predictor p, y(k) is the k^{th} symbol of the y time series.

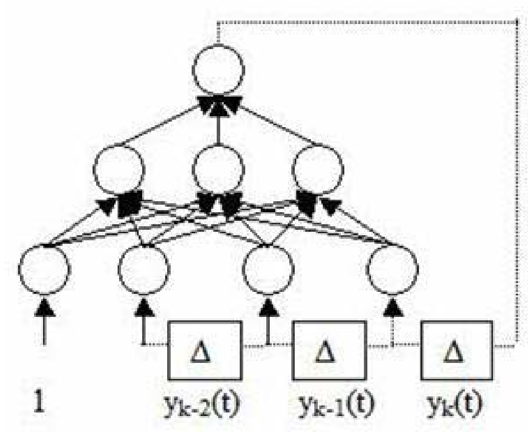

Figure 4. Recurrent neural network

This equation becomes in the multi (n) step-ahead prediction:

$$\hat{y}(k+n) = p[\hat{y}(k+n-1), \quad \hat{y}(k+n-2), ..., \hat{y}(k+n-N)] \qquad (2)$$

Multi-step-ahead prediction (n step-ahead) is the concatenation of n single-step-ahead prediction. We perform single-step-ahead prediction by a recurrent neural network with N input neurons and a single output neuron. The concatenation of many predictors to build a Multi-step-ahead predictor is accomplished by connecting the output of k^{th} network to the input of the $(k+1)^{th}$ network. This concatenation is equivalent to a single recurrent neural network with 2n hidden layers. Multi-step-ahead prediction is of particular importance in large hierarchical complex systems as single step prediction can not be propagated fast enough through the hierarchy to allow efficient control.

Learning algorithm

The learning algorithm is based on the minimization of the cost function J given by the following equation:

$$J = \frac{1}{2}\sum_{i=1}^{n}[\hat{y}\ (k+i) - y^{d}\ (k+i)]^2 = \frac{1}{2}\sum_{i=1}^{n} Ai \tag{3}$$

With $\hat{y}(k+i)$ is the estimated output by the recurrent neural network, $y^d(k+i)$ is the desired one, n is the length of the prediction window. The derivative of the cost function with respect to a connection weight parameter (θ) is given by:

$$\frac{\partial J}{\partial \theta} = \frac{1}{2}\sum_{i}\frac{\partial Ai}{\partial \hat{y}\ (k+i)} \cdot \frac{\partial \hat{y}\ (k+i)}{\partial \theta} \tag{4}$$

$$\frac{\partial J}{\partial \theta} = \sum_{i}[\hat{y}\ (k+i) - y^{d}(k+i)] \cdot \left[\sum_{j=1}^{N}\frac{\partial \hat{y}\ (k+i)}{\partial \hat{y}\ (k+i-j)} \cdot \frac{\partial \hat{y}\ (k+i-j)}{\partial \theta} + \frac{\partial \hat{y}\ (k+i)}{\partial \theta}\right] \tag{5}$$

With N is the number of recurrent input in the first layer, θ is a connection weight parameter of the recurrent neural network. The weights correction of the recurrent neural network is executed at the end of each prediction window according to:

$$\theta_{new} = \theta_{old} - \varepsilon \cdot \frac{\partial J}{\partial \theta} \tag{6}$$

Recurrent neural network training is recognized as very delicate procedure especially for an important prediction window length (multi-step ahead). The nature of the predicted data has a great influence in the convergence speed of the training phase. The above RNN model cannot handle large and noisy data sets as might appear in large hierarchical complex systems. We need a multiple recurrent neural network structure allowing the prediction of noisy and large data sets.

Multiple recurrent neural networks for prediction

This section describes the synthesis phases of a multiple recurrent neural network predictor composed by a SOM (Self Organizing Map) combined with a set of local RNN predictors [Chtourou 2006]. At each sampling time, the proposed architecture provides the next n components of the time series. The proposed prediction approach is based on two stages: the classification and the prediction. The classification process arranges the data set into many sub-classes using a one dimensional SOM. Each sub-class is used to train its associated local RNN. Each SOM output neuron corresponds to a local RNN.
The prediction ability is deeply dependent on the RNN architecture. The recurrent neural network architecture is determined by the width of the input and hidden layers. In the presented study, the width of the input vector has been fixed based on several experiences. However, the hidden neurons number has been determined following an incremental training procedure .

Preliminary Evaluation

We have used the described neural network model as a predictor on a hypothetical hierarchical complex system. According to our experiment a neural network predictor is attached to each system in order to predict the system activity.

We will consider the three cases described figure 2:

1. **case 1:** independent prediction of systems in order to assess the quality of the predictors. In this case the inputs of the systems are random and therefore exhibit varying activity which require monitoring and prediction.

2. **case 2:** some systems are stable with coupled prediction: in this case the predictors are hierarchically connected with some systems having stable activity therefore with no specific needs for prediction.

3. **case 3:** stable systems are not connected. The overall system activity prediction does not take into account stable systems and no predictor is assigned to these systems. The overall system activity is based on the hierarchically connected predictors of the varying units.

Figure 5 and 6 illustrates case 1, figure 7 and 8 case 2 and finally figure 9 and 10 case 3.

Case 1: All sub-systems have some activity.

In this case we will vary all sub-systems and have assigned activity monitoring units predict activity and transmit this prediction to upper level systems. Upper level systems combine all incoming predictions to generate predictions on the related system activity. The following figure shows for the sub-systems both their measured and predicted activity. Clearly the measured and predicted activity is close and provide a useful tool to control the activity of the system. However, this case comes at the cost of assigning a prediction unit to each subsystem.

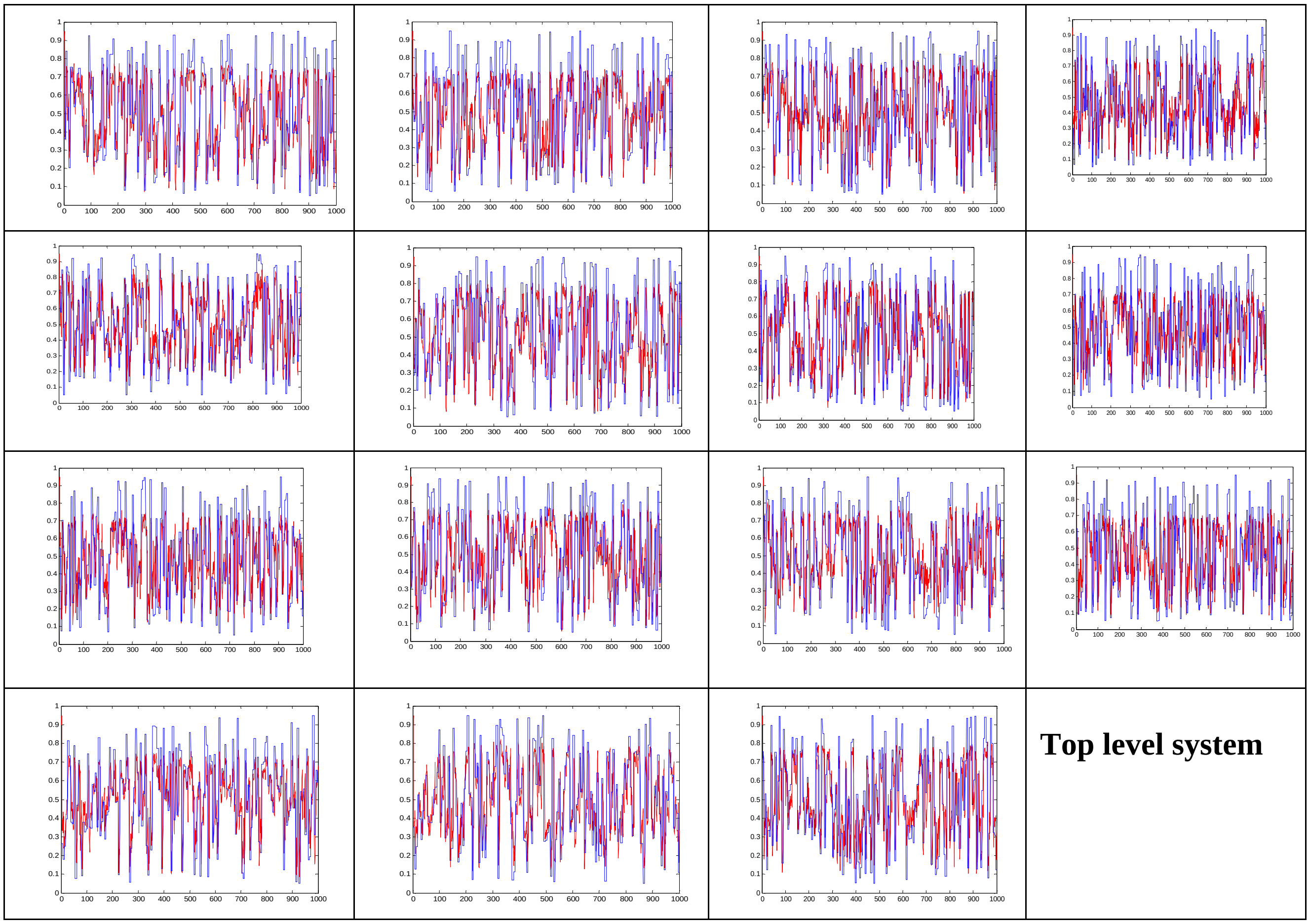

Figure 5. Case 1: measured and predicted activity for each sub-system at each level of the hierarchy with random inputs in all sub-systems

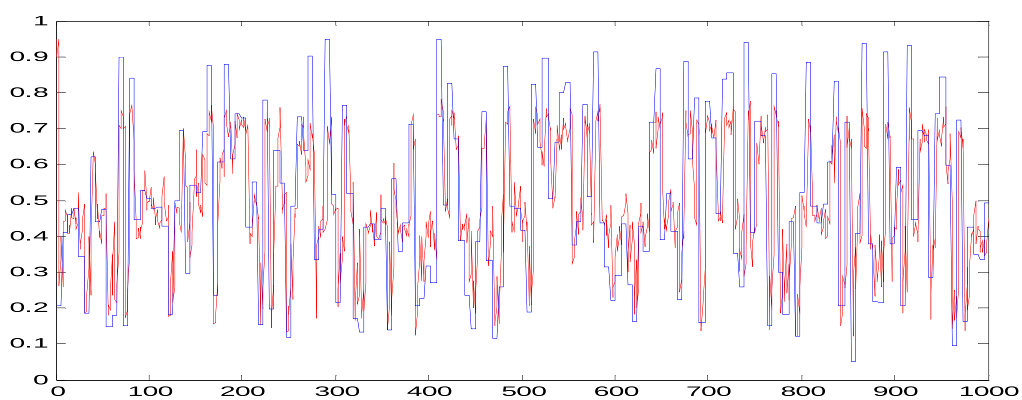

Figure 6. Comparison of Global System Activity Measured and Predicted

Case 2: Some activity inputs are varying with some systems activity stable.

In this case we will vary all sub-systems except a randomly selected subset in the hierarchy of systems which will have very stable activity and have assigned activity monitoring units predict activity and transmit this prediction to upper level systems. Upper level systems again combine all incoming predictions to generate predictions on the related system activity. The following figure shows for the sub-systems both their measured and predicted activity. Clearly the measured and predicted activity is close and provide a useful tool to control the activity of the system. However, still this case comes at the cost of assigning a prediction unit to each subsystem even if some systems have a stable activity and therefore do not need special prediction effort and resource.

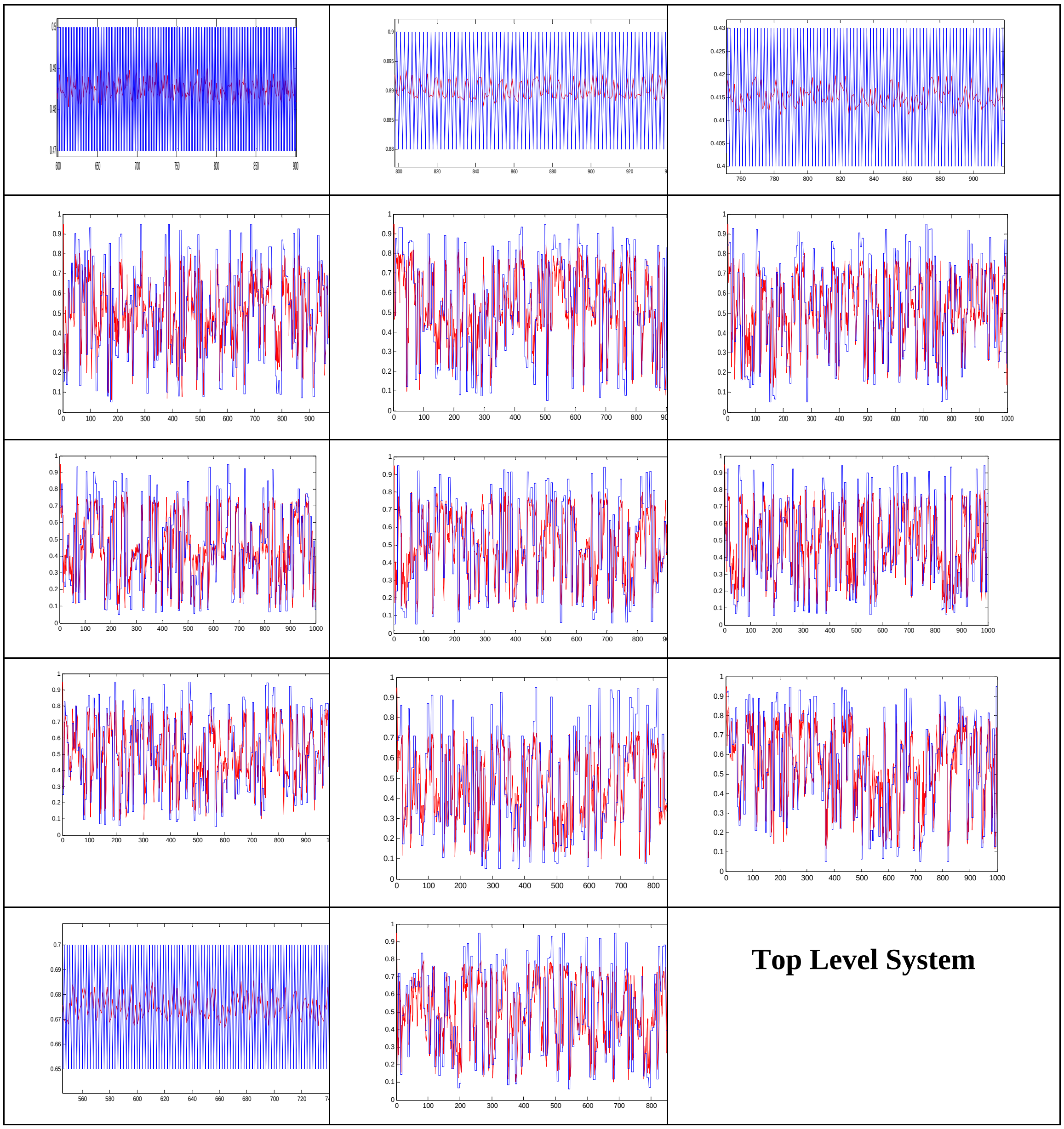

Figure 7. Case 2: measured and predicted activity for each sub-system at each level of the hierarchy with random inputs in all sub-systems except some subsystems with little activity variation

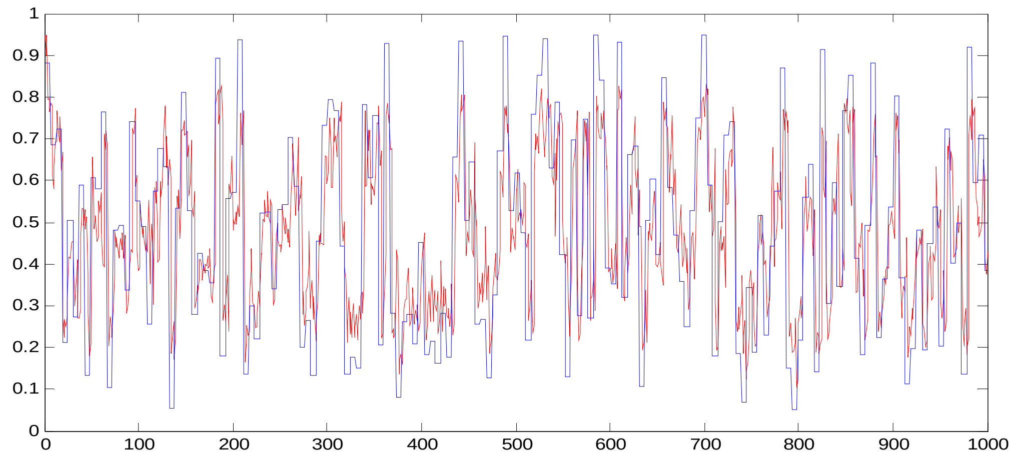

Figure 8. Comparison of Global System Activity Measured and Predicted

Case 3: Some activity inputs are varying

In this case similar to case 2 we will vary all sub-systems except a randomly selected subset in the hierarchy of systems which will have very stable activity however we will have assigned activity monitoring units predict activity only on varying unit and transmit this prediction to upper level systems. Upper level systems again combine all incoming predictions to generate predictions on the related system activity. The following figure shows for the sub-systems both their measured and predicted activity.

Figure 8 describes some samples of systems prediction at different levels of the hierarchy while ignoring stable activity systems. Figure 9 gives the final prediction result of the top level system while ignoring the systems exhibiting stable behavior. As it can be observed the overall predicition is good and allows a good estimate of an hierarchical system activity in presence of some stable systems. This results allows system designer to focus of   more evolving part of the system in order to adapt the system.

This study come as a first step towards support to new research avenues on engineering systems of systems [Sage 2007][Lewis2008][Rebovich 2008][Simpson 2008] through profiling [Stevens 2008] with the objective of continuous allocation of the right amount of resource [Wang 2007].

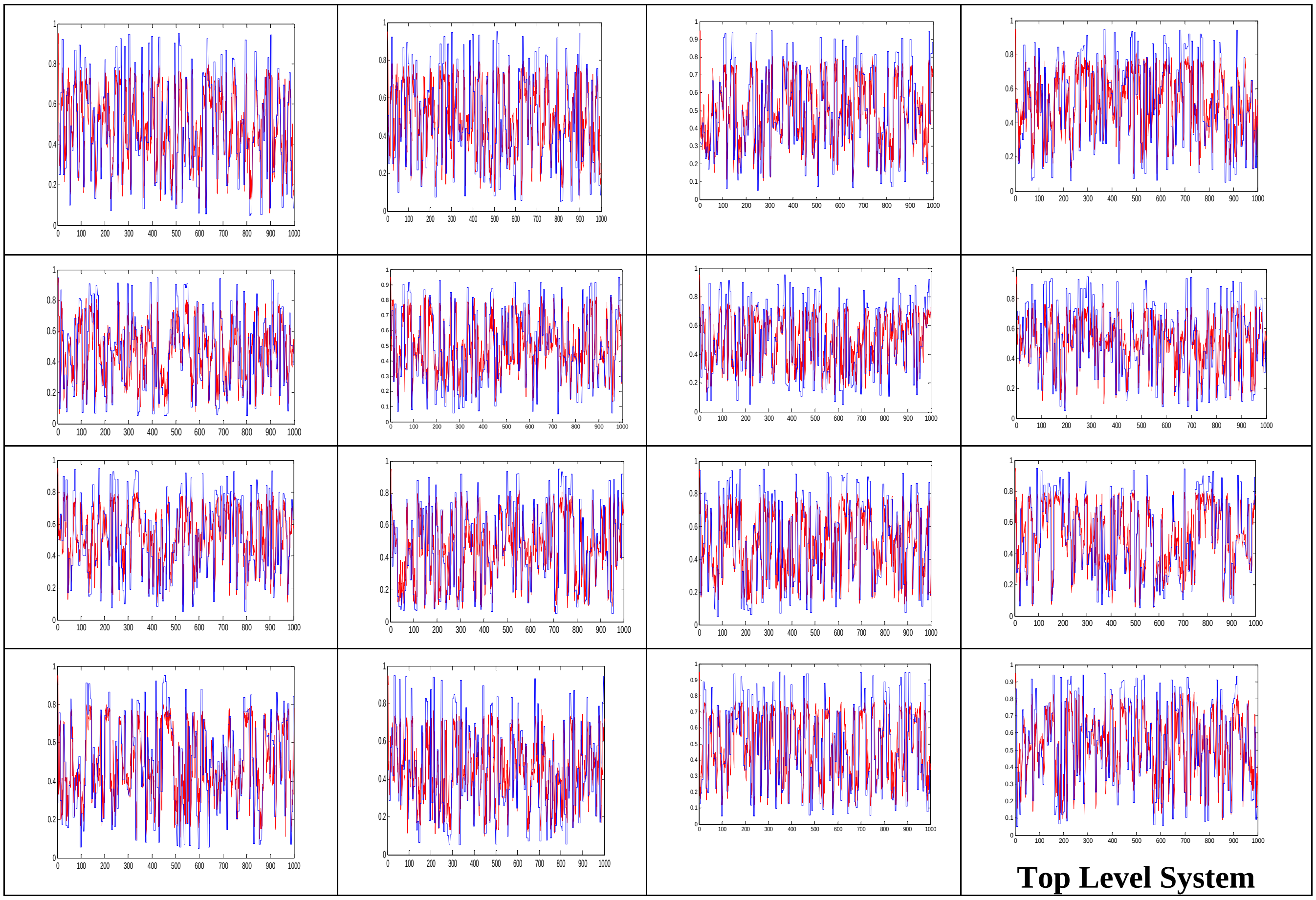

Figure 9. Case 3: measured and predicted activity for each sub-system at each level of the hierarchy and ignoring sub-systems with little activity variation

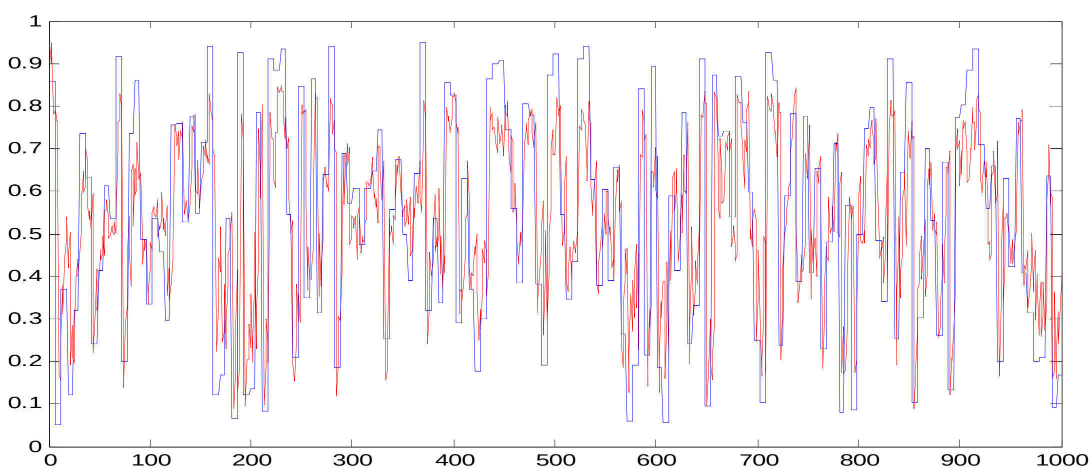

Figure 10. Global result - Hierarchical prediction with ignoring inputs in some sub-systems

Conclusion

Complex systems composed of multi-level hierarchical systems exhibit complex interacting dependences which affects the performance of the overall system. Although, one solution is to tightly monitor each system activity and feedback to the next step level in order to take appropriate measures and actions to improve the system this solution is costly and too systematic. We described in this paper an activity driven adaptive prediction technique for complex systems which minimizes monitoring and prediction resources requirements and still keep efficient overall systems performance prediction. This was achieved through neural network techniques which present the ability to learn dynamic complex behavior. This work is a preliminary work for the establishment of a general framework for adaptive complex systems.

We plan to improve our model and apply this framework for actual systems and system of systems with evolving requirements.

Acknowledgement

The author would like to thank all the research team members for their support.

References

Atiya, Amir F. and Parlos, Alexander G. 2000. "New Results on Recurrent Network Training: Unifying the Algorithms and Accelerating Convergence", IEEE Transaction on Neural Networks, Vol. 11, No. 3, pp. 697-709.

Barr J. and Saraceno. 2008. F. Organization, Learning and Cooperation, Journal of Economic Behavior & Organization.

Bjelkemyr, M., Semere, D. , Lindberg, B. 2007. An Engineering Systems Perspective on System of Systems Methodology, IEEE International Systems Conference, Waikiki Beach, Honolulu, Hawaii, USA.

Bode, J., 1998. Decision support with neural networks in the management of research and development: Concepts and application to cost estimation. Information & Management 34 33-40.

Breitler A.L.and Sloan C.D., 2004. Predicting the Reliability of a Complex System Using an Artificial Neural Network, INCOSE 2004 - 16th Annual International Symposium Proceedings.

Chtourou, S. Chtourou, M. Hammami, O. 2006. "Neural Network Based Memory access Prediction Support for Soc Dynamic Reconfiguration", Proceedings of the International Joint Conference Neural Network, pp. 5130-5136.

Connor T. and Douglas, R. 1994. "Recurrent Neural Networks and Robust Time Series Prediction", IEEE Transactions on Neural Networks, Vol. 5, No.2, pp. 240-254.

Corsello M.A. 2008. System-of-Systems Architectural Considerations for Complex Environments and Evolving Requirments, IEEE Systems Journal, Vol. 2, No. 3, Sept.pp. 312-320.

DeRosa, J.K., Grisogno, A., Ryan A.J., and Norman, D.O.A. 2008.Research Agenda for the Engineering of Complex Systems, IEEE International Systems Conference, Montreal, Canada.

DeRosa, J.K.2007.Combined Systems Engineering and Management in the Evolution of Complex Adaptive Systems, IEEE International Systems Conference, Waikiki Beach, Honolulu, Hawaii, USA.

Elm, J.P. 2008. A Study of Systems Engineering Effectiveness – Initial Results, IEEE International Systems Conference, Montreal, Canada.

Lewis, G. Morris, E. Place, P. Simanta, S. Smith D. and Wrage L.2008. Engineering Systems of Systems, IEEE International Systems Conference, Montreal, Canada.

Lin, T. Horn, G., Tino P. and Lee Giles, C. 1996. “Learning Long-term Dependencies in NARX Recurrent Neural Networks”, IEEE Transactions on Neural Networks, Vol. 7, No. 6,pp. 1329.

Lin, T., Giles, C. L, Horne, B. Kung, S.Y. , 1997. “A Delay Damage Model Selection Algorithm for NARX Neural Networks“, IEEE Transaction on Signal Processing, Vol. 45, No. 11, pp. 2719-2730.

Nerrand, O. Roussel-Ragot, P. Urbani, D. Personnaz L. and Dreyfus, G. 1994. “Training Recurrent Neural Networks: Why and How? An illustration in Dynamical Process Modeling”, IEEE Transactions on Neural Networks, Vol. 5, No.2, pp. 178-184.

On Cheung, S., Shek Pui Wong, P. , Fung, A. S. Y. and Coffey, W.V. 2006. Predicting project performance through neural networks, International Journal of Project Management 24 207-215.

Parlos, A. G., Rais, O.T. Atiya, A.F. 2000. “Multi-step-ahead Prediction Using Dynamic Recurrent Neural Networks”, Neural Networks 13, pp. 765-786.

Rebovich, G. 2008. The Evolution of Systems Engineering, IEEE International Systems Conference, Montreal, Canada.

Sage, A.P. and Biemer S.M., 2007. Processes for System Family Architecting, Design and Integration, IEEE Systems Journal, Vol. 2, No. 3, pp. 5-16.

Scully, J.K.2008. The Role of System Behavior in Diagnostic Performance, IEEE International Systems Conference, Montreal, Canada.

Selby R.W. 2006. Generating Predictive Models Using Decision Trees and Neural Networks for Large Scale Systems Engineering, INCOSE 2006 - 16th Annual International Symposium Proceedings.

Simpson J.J. and Dagli, C.H. 2008. System of Systems Architecture Generation and Evaluation using Evolutionary Algorithms, IEEE International Systems Conference, Montreal, Canada.

Stevens, R. 2008. Profiling Complex Systems, IEEE International Systems Conference, Montreal, Canada.

Wang, L. Fang L. and Hipel, K.W. 2007. On Achieving Fairness in the Allocation of Scarce Resources: Measurable Principles and Multiple Objective Optimization Approaches, IEEE Systems Journal, Vol. 2, No. 3, pp. 17-28.

Wells, B.H. 2008. A Multi-dimensionnal Hierarchical Engineering Competency Model Framework, IEEE International Systems Conference, Montreal, Canada.

William Chien, T. Lin , C. Tan B.and Chuan Lee, W. 1999. A neural networks-based approach for strategic planning, Information & Management 35, 357-364.

BIOGRAPHY

Omar Hammami is an Associate Professor with ENSTA/DGA since 2000. Prior to that he was Associate Professor in the University of Aizu , Japan and Head of the Performance Evaluation Laboratory. He received his Phd in Computer Science from University of Toulouse and have been an assistant professor with ENSEEIHT, Toulouse. Omar Hammami have been involved in numerous R&D projects at national and international levels in embedded systems design methodologies, microelectronics and high performance computing. His current interets lies in Cognitive radio, Systems Engineering and Network science.