HiNFRA: Hierarchical Neuro-Fuzzy Learning for Online Risk Assessment

Kjetil Haslum, Ajith Abraham and Svein Knapskog
Center for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology
O.S. Bragstads plass 2E, N-7491 Trondheim, Norway

E-mail: {haslum, ajith.abraham, knapskog}@q2s.ntnu.no

Abstract

Our previous research illustrated the design of fuzzy logic based online risk assessment for Distributed Intrusion Prediction and Prevention Systems (DIPPS) [3]. Based on the DIPPS sensors, instead of merely preventing the attackers or blocking traffic, we propose a fuzzy logic based online risk assessment scheme. This paper propose a Hierarchical Neuro-Fuzzy online Risk Assessment (HiNFRA) model to aid the decision making process of a DIPPS. The fine tuning of fuzzy logic based risk assessment model is achieved using a neural network learning technique. Preliminary results indicate that the neural learning technique could improve the fuzzy controller performance and make the risk assessment model more robust.

1. Introduction

Intrusion prevention systems are proactive defense mechanisms designed to detect malicious packets embedded in normal network traffic and stop intrusions dead, blocking the offending traffic automatically before it does any damage rather than simply raising an alert as, or after, the malicious payload has been delivered. DIPPS are simply a superset of the conventional Intrusion Prevention System (IPS) implemented in a distributed environment. Basic architecture of a DIPPS element is depicted in Figure 1. We consider IPS as an integrated Intrusion Detection System (IDS) with many additional functions. Due to the distributed nature of IPS, the implementation poses several challenges. The IDSs are embedded inside software mobile agents and placed in the network to be monitored [1]. An individual IDS may be configured to detect a single attack, or it may detect several types of attacks.

In a large network, each DIPPS element communicates/coordinates with other DIPPS local controllers and/or a central controller. The Hidden Markov Model (HMM) [7] model processes the attack data information from the various mobile agent IDS sensors [6]. Based on the nature of the detected attack, the following actions would be taken [2]:

1. If the detected attack is simply a port scan or a probe, the HMM model attempts to make a prediction of a possible future attack based on the current distributed attack patterns. Based on this prediction, the central controller (or administrator) would take precautionary measures to prevent future attacks. The central controller would also make use of an online risk assessment of the assets subjected to this possible serious attack in the future.

2. If the detected attack is very serious, the central controller would take necessary actions to re-configure firewall rules or notify the administrator etc. Such serious attacks would bypass the HMM model.

3. At any time any abnormal traffic rate is noted by the monitor if a predetermined level is reached, the central controller may take necessary actions to re-configure firewall rules or notify the administrator etc.

Risk assessment is often done by human experts, because there is no exact and mathematical solution to the problem. Usually the human reasoning and perception process cannot be expressed precisely. Different people have different opinions about risk and the association of its dependent variables, and fuzzy logic provides an excellent framework to model this [3].

A Fuzzy Inference System (FIS) [9] can utilize human expertise by storing its essential components in rule base and database, and perform fuzzy reasoning to infer the overall output value. The derivation of *if-then* rules and corresponding membership functions depends heavily on the *a priori* knowledge about the system under consideration. However there is no systematic way to transform experiences of knowledge of human experts to the knowledge base

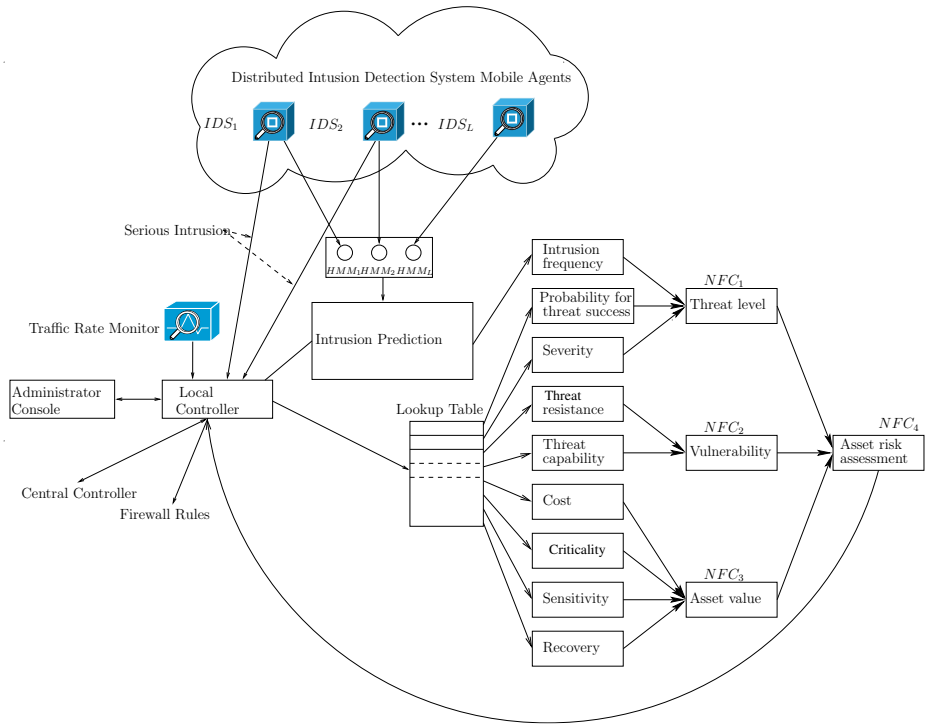

Figure 1. Architecture of a DIPPS element

of a FIS. There is also a need for adaptability or some learning algorithms to produce outputs within the required error rate.

This paper is focused on the development of neural network learning techniques for the optimization of a fuzzy risk assessment system. The rest of this paper is organized as follows. Section 2 presents the proposed neuro-fuzzy risk assessment model. Experiment results are given in Section 3 followed by conclusions towards the end.

2. Neuro-Fuzzy Risk Assessment Model

2.1 Fuzzy modeling of risk

For risk assessment, nine basic linguistic variables are used that are processed using three Neuro-Fuzzy Controllers ($NFC_1 - NFC_3$). The three NFC's represent *Threat Level*, *Vulnerability* and *Asset Value*, which are three derived linguistic variables. Threat level is modeled using three linguistic variables: *intrusion frequency*, *probability of threat success* and *severity*. We model vulnerability as a derived variable from *threat resistance* and *threat capability*. The asset value is derived from three linguistic variables: *Cost*, *Criticality*, *Sensitivity* and *Recovery*. The derived linguistic variables are then combined using NFC_4 to compute the net *Asset Risk*. This forms a hierarchical fuzzy system as shown in Figure 2.

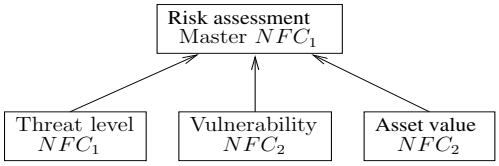

Figure 2. Hierarchical architecture of four fuzzy logic controllers

Values for the input variables are estimated based on the information from the HMM module, the DIDS and the traffic rate monitor. To simplify the calculation of input values, we have used the same attack categories as proposed by MIT Lincoln Laboratory - DARPA IDS evaluation datasets IDS [5]. The local controller uses information from the DIDS and the traffic rate monitor to predict which attack category the next attack will fit into.

2.2 Fuzzy risk model optimization using neural learning

In an integrated model, neural network learning algorithms are used to determine the parameters of fuzzy in-

ference systems. Integrated neuro-fuzzy systems share data structures and knowledge representations. A fuzzy inference system can utilize human expertise by storing its essential components in rule base and database, and perform fuzzy reasoning to infer the overall output value. The derivation of *if-then* rules and corresponding membership functions depends heavily on the *a priori* knowledge about the system under consideration. However there is no systematic way to transform experiences of knowledge of human experts to the knowledge base of a fuzzy inference system. There is also a need for adaptability or some learning algorithms to produce outputs within the required error rate. On the other hand, Artificial Neural Network (ANN) learning mechanism does not rely on human expertise. Due to the homogenous structure of ANN, it is hard to extract structured knowledge from either the weights or the configuration of the network. The weights of the neural network represent the coefficients of the hyper-plane that partition the input space into two regions with different output values. If we can visualize this hyper-plane structure from the training data then the subsequent learning procedures in a neural network can be reduced. However, in reality, the a priori knowledge is usually obtained from human experts, it is most appropriate to express the knowledge as a set of fuzzy if-then rules, and it is very difficult to encode into a neural network.

To a large extent, the drawbacks pertaining to these two approaches seem complementary. Therefore, it seems natural to consider building an integrated system combining the concepts of FIS and ANN modeling. A common way to apply a learning algorithm to a fuzzy system is to represent it in a special neural network like architecture. However the conventional neural network learning algorithms (gradient descent) cannot be applied directly to such a system as the functions used in the inference process are usually non differentiable. This problem can be tackled by using differentiable functions in the inference system or by not using the standard neural learning algorithm. In our simulation, we used the Adaptive Network Based Fuzzy Inference System (ANFIS) [4]. ANFIS implements a Takagi Sugeno Kang (TSK) fuzzy inference system [8] in which the conclusion of a fuzzy rule is constituted by a weighted linear combination of the crisp inputs rather than a fuzzy set.

For a first order TSK model, a common rule set with two fuzzy *if-then* rules is represented as follows:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$

Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$

where x and y are linguistic variables and A_1, $A_{2,}$, $B_{1,}$, B_2 are corresponding fuzzy sets and p_1 , q_1, $r_1$and p_2, q_2, r_2 ,are linear parameters.

ANFIS makes use of a mixture of back propagation to learn the premise parameters and least mean square estimation to determine the consequent parameters. A step in the learning procedure has two parts: In the first part the input patterns are propagated, and the optimal conclusion parameters are estimated by an iterative least mean square procedure, while the antecedent parameters (membership functions) are assumed to be fixed for the current cycle through the training set. In the second part the patterns are propagated again, and in this epoch, back propagation is used to modify the antecedent parameters, while the conclusion parameters remain fixed. This procedure is then iterated.

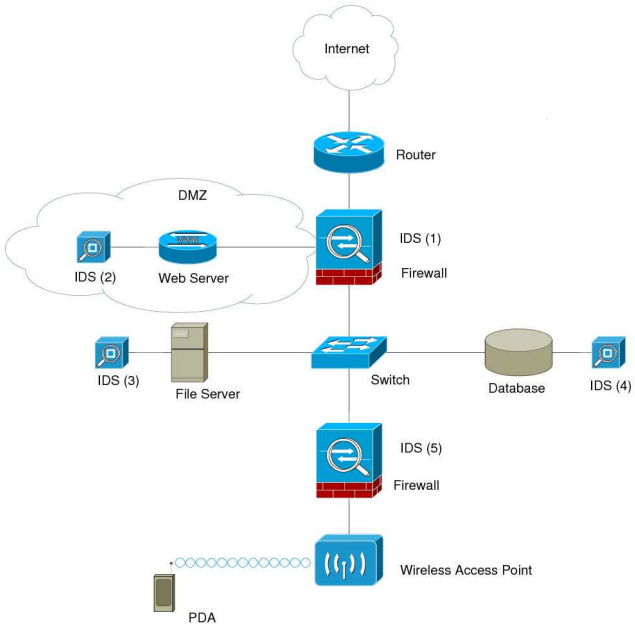

Figure 3. Example network showing assets and IDS agents

3. Experiment Results

In order to illustrate the neuro-fuzzy risk assessment model, we constructed a small network model as illustrated in Figure 3. The sample network consists of four different assets; a router, a public web server, a file server, and a database. Five IDS Agents denoted by $IDS_1, \ldots, IDS_5$ are deployed in the network, and the observations are sent to the their corresponding HMM. The attack category used for the risk assessment is based on inputs from the IDS agents and this value is used to assign values to eight of the nine input variables. Only the Intrusion Frequency is estimated based on the output from the HMM module.

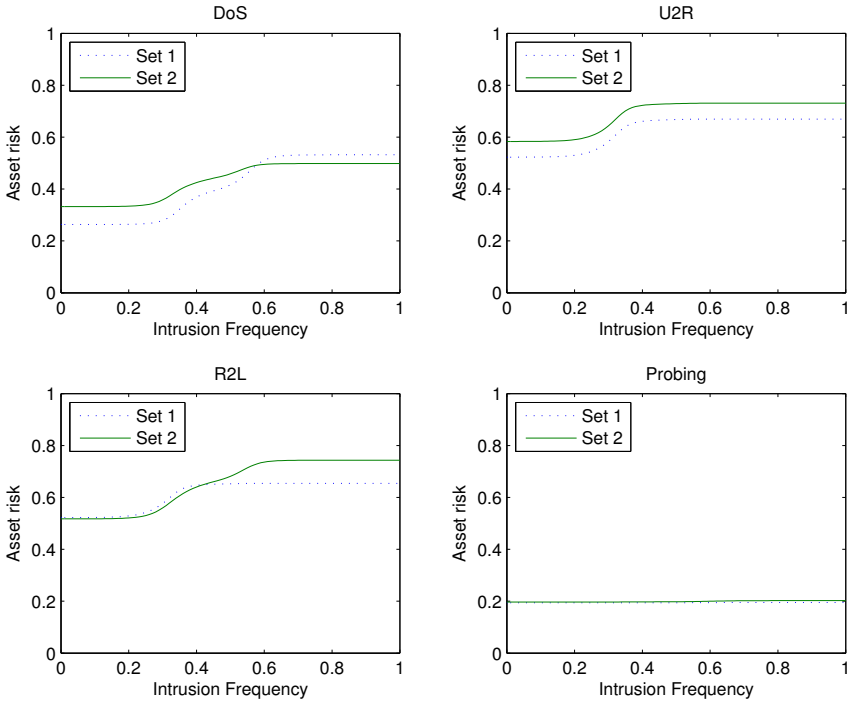

Figure 4. Parameter sensitivity for different attack categories

Table 1. Learning results for NFC1

MF		tri	gbell	gauss	psig
Data set 1	#MF	2	2	2	2
	Epochs	49	27	17	28
	Test error	0.0969	0.1159	0.1002	0.0881
	Train error	0.0203	0.0172	0.0192	0.0211
Data set 2	#MF	2	2	2	2
	Epochs	45	37	22	8
	Test error	0.1371	0.1500	0.1091	0.1732
	Train error	0.0220	0.0225	0.0228	0.0480
Data set 3	#MF	2	2	2	2
	Epochs	42	14	16	4
	Test error	0.1239	0.0694	0.0756	0.0513
	Train error	0.0251	0.0379	0.0379	0.0303

Table 2. Learning results for NFC2

MF		tri	gbell	gauss	psig
Data set 1	#MF	2	3	3	3
	Epochs	22	1	1	1
	Test error	0.0306	0.0392	0.0368	0.0444
	Train error	0.0400	0.0256	0.0252	0.0256
Data set 2	#MF	3	2	2	3
	Epochs	1	1	2	1
	Test error	0.0473	0.0443	0.0568	0.0528
	Train error	0.0214	0.0681	0.0294	0.0184
Data set 3	#MF	3	3	3	3
	Epochs	1	1	1	1
	Test error	0.0223	0.0350	0.0348	0.0376
	Train error	0.0308	0.0272	0.0267	0.0273

3.1 Hierarchical neuro-fuzzy modeling

To avoid any bias in the learning process, we randomly sampled three sets of data from the master data set. 75 % of the data was used for training and the remaining for test data. We implemented a Hierarchical Neuro-Fuzzy Risk Assessment (HiNFRA) model as illustrated in Figure 2. The performance of the four controllers for different membership functions for three different data sets are depicted in Tables 1, 2, 3 and 4. We used four different Membership Functions (MF): Triangular (tri), Gaussian bell (gbell), Gaussian (gauss) and product of two sigmoidal function (psig). As evident, depending on the membership function and data set used, different controllers are using varying number of epochs. In some cases, the NFC was constructed after 1 learning epoch. The four NFC's were build individually and then connected as shown in Figure 2.

The developed HiNFRA model is then tested using two

Table 3. Learning results for NFC3

MF		tri	gbell	gauss	psig
Data set 1	#MF	2	2	2	2
	Epochs	1	8	1	22
	Test error	0.0840	0.0726	0.0686	0.0788
	Train error	0.0342	0.0536	0.0498	0.0557
Data set 2	#MF	2	2	2	2
	Epochs	1	1	1	68
	Test error	0.0627	0.1187	0.0665	0.1339
	Train error	0.0352	0.0552	0.0470	0.0290
Data set 3	#MF	2	2	2	2
	Epochs	1	20	14	30
	Test error	0.2268	0.2206	0.2644	0.1170
	Train error	0.0332	0.0344	0.0328	0.0423

Table 4. Learning results for NFC4

MF		tri	gbell	gauss	psig
Data set 1	#MF	2	2	2	2
	Epochs	2	7	11	4
	Test error	0.1691	0.1488	0.1555	0.1379
	Train error	0.0748	0.0829	0.0724	0.1031
Data set 2	#MF	2	2	2	2
	Epochs	51	17	15	19
	Testing error	0.1541	0.1557	0.1689	0.2250
	Training error	0.0649	0.0603	0.0600	0.0600
Data set 3	#MF	2	2	2	2
	Epochs	39	20	20	24
	Testing error	0.1396	0.1122	0.1148	0.1098
	Training error	0.0561	0.0508	0.0485	0.0558

Table 5. Lookup Table1

	Attack Categories			
Variable	DoS	U2R	R2L	PR
Intrusion frequency	0.25	0.25	0.25	0.25
Pr threat success	0.90	0.70	0.70	0.10
Severity	0.40	0.90	0.90	0.30
Threat level	0.38	0.52	0.52	0.29
Threat resistance	0.10	0.60	0.90	0.20
Threat capabilit	0.50	0.85	0.80	0.10
Vulnerability	0.46	0.36	0.15	0.10
Cost	0.30	0.30	0.30	0.30
Criticality	0.70	0.70	0.70	0.10
Sensitivity	0.15	0.85	0.85	0.20
Recovery	0.40	0.85	0.70	0.15
Asset value	0.33	0.52	0.52	0.24
Asset risk	**0.27**	**0.54**	**0.54**	**0.19**

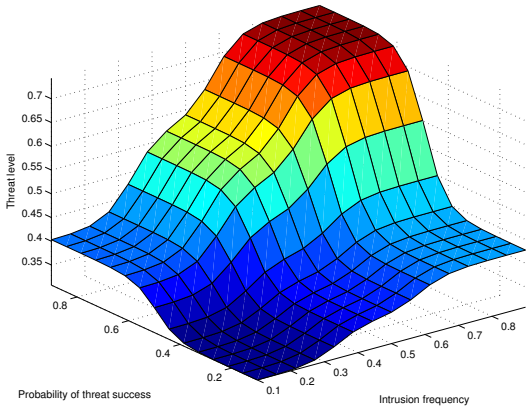

Figure 5. Control Surface View of NFC_1

sets of data based on an attack situation. Specific information about different attack categories are stored in a lookup table. All values in the lookup table is scaled within the range $0 - 1$. The attack category used for the risk assessment is based on inputs from the IDS agents and this value is used to assign values to eight of the nine input variables. Only the Intrusion Frequency is estimated based on the output from the HMM module.

The two lookup tables and final results (asset risk) are illustrated in Tables 5 and 6.

Figure 4 illustrates the asset risk values for different intrusion frequency variations (0-1). For the different parameter settings (Tables 5 and 6), as evident from Figure 4, the asset risk values show clear sensitivity for each attack category. This also illustrates that the proposed system is very adaptive for different attack categories under varying conditions. Figures 5 - 8 illustrates the surface plots of the four developed controllers.

4. Conclusions

This paper presented a detailed implementation of the Hierarchical Neuro-Fuzzy online Risk Assessment (HiNFRA) model to aid the decision making process of DIPPS. The fine tuning of fuzzy logic based risk assessment model is achieved using neural network learning technique. Preliminary results indicate that neural learning techniques could improve the fuzzy controller performance and make the risk assessment model more robust. Compared to our previous model [3], where the fuzzy *if-then* rules were formulated based on expert knowledge, the implementation of HiNFRA is more simple and adaptive.

Our future research is targeted to further develop and op-

Table 6. Lookup Table2

	Attack Categories			
Variable	DoS	U2R	R2L	PR
Intrusion frequency	0.25	0.25	0.25	0.25
Pr threat success	0.70	0.70	0.50	0.10
Severity	0.50	0.90	0.70	0.45
Threat level	0.42	0.52	0.44	0.31
Threat resistance	0.20	0.80	0.70	0.20
Threat capabilit	0.40	0.80	0.80	0.10
Vulnerability	0.33	0.20	0.26	0.10
Cost	0.40	0.40	0.50	0.30
Criticality	0.60	0.80	0.80	0.10
Sensitivity	0.20	0.80	0.70	0.10
Recovery	0.30	0.80	0.50	0.25
Asset value	0.39	0.59	0.60	0.24
Asset risk	**0.39**	**0.60**	**0.53**	**0.20**

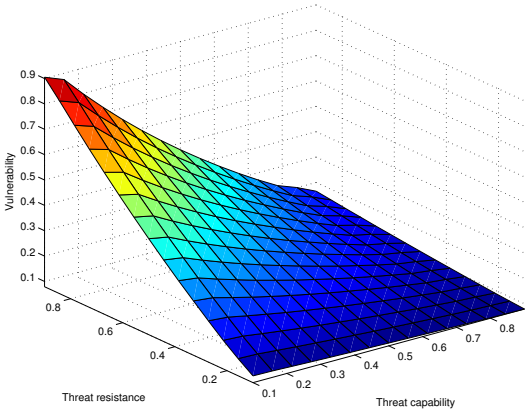

Figure 6. Control Surface View of NFC_2

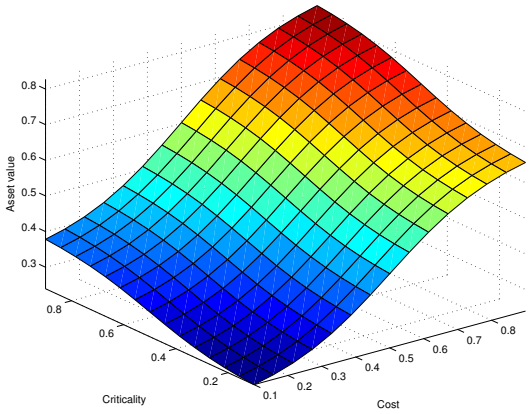

Figure 7. Control Surface View of NFC_3

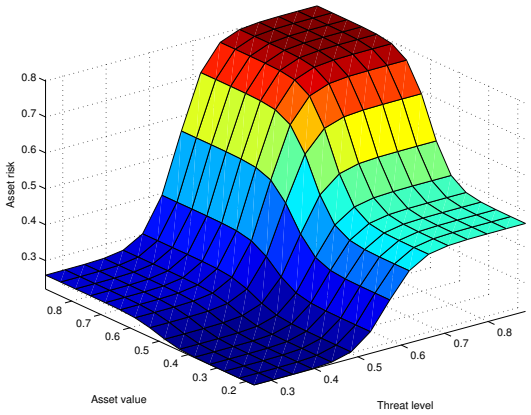

Figure 8. Control Surface View of NFC_4

timize fuzzy risk models using evolutionary algorithms.

References

[1] A. Abraham, R. Jain, J. Thomas, and S. Han. D-scids: Distributed soft computing intrusion detection systems. *Journal of Network and Computer Applications, Elsevier Science*, 30(1):81–98, 2007.

[2] K. Haslum, A. Abraham, and S. Knapskog. Dips: A framework for distributed intrusion prediction and prevention using hidden markov models and online fuzzy risk assessment. In *Third International Symposium on Information Assurance and Security, IEEE Computer Society press*, volume I, pages 183–188, 2007.

[3] K. Haslum, A. Abraham, and S. Knapskog. Fuzzy online risk assessment for distributed intrusion prediction and prevention systems. In *Tenth International Conference on Modeling and Simulation, IEEE Computer Society press*, volume I, page in press, 2008.

[4] J.-S. Jang. Anfis: adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(3):665–685, May/Jun 1993.

[5] K. Kendall. A database of computer attacks for the evaluation of intrusion detection systems. Master's thesis, MIT, USA, June 1999.

[6] R. Khanna and H. Liu. System approach to intrusion detection using hidden markov model. In *IWCMC '06: Proceeding of the 2006 international conference on Communications and mobile computing*, pages 349–354, New York, NY, USA, 2006. ACM Press.

[7] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, pages 267–296, 1990.

[8] M. Sugeno. *Industrial Applications of Fuzzy Control*. Elsevier Science Inc., New York, NY, USA, 1985.

[9] L. Zadeh. Fuzzy sets. *Info. & Ctl.*, 8:338–353, 1965.