UDC 681.518

A. MILANI FARD, V. SALMANI, M. H. ANSARI, M. NAGHIBZADEH, A. FARZAD, M. DELGIR, A. ZAMIRI, H. KAMYAR, E. SABOORI, F. SEIFI, M. MOZAFARI,
Software Simulation and Modeling Laboratory, Faculty of Engineering, Ferdowsi University of Mashhad, IRAN,
nexus@ferdowsi.um.ac.ir

NEXUS ROBOSOCCER DEVELOPMENT PROCESS FROM 2D SIMULATED AGENTS TO 3D HUMANOID SOCCERBOTS

Soccer simulation as an effort for motivating researchers in the field of artificial intelligence and robotic research has always been a progressive approach. Robotic soccer is a particularly good domain for studying multi-agent systems and behaviors. In this paper, we describe researches done by Nexus team from the prior 2D soccer simulation environment to the curent humanoid simulation version. The main development features were done on decision making, action selection, and coach strategy making modules using fuzzy logic mechanism and game theory approach. Some very basic humanoid actions are also explained.

1. Introduction. Robotic soccer is a particularly good domain for studying multi-agent systems. It has been gaining popularity in recent years with international competitions like RoboCup which is planned for the near future [1]. Soccer simulation environment is a client-server platform which provides an excellent testbed for development of multi-agent systems. Using this testbed frees researchers fomr getting involved in the complexities of physical robot developmets.
Nexus[1], established in 2002, is the RoboCup soccer simulation team of Ferdowsi University of Mashhad, Iran. As an important first step, the team was qualified to participate in RoboCup contest in 2003 Padova, Italy, in Soccer-2D league. Afterwards, Nexus could go as high as the third round in RoboCup 2005 Osaka, Japan, and ranked between 9th and 12 th place among 33 teams. The team has also participated and won some domestic leagues in Iran during these times. Currently, Nexus has become eligible and was qualified to participate in the RoboCup 2007 Atlanta, US, in the humanoid soccer robot simulation league. In this paper, we propose a comprehensive review of our research projects done in the RoboCup simulation filed from the early stablishement of the Nexus team.

2. Applied Techniques to 2D Soccer Agents. One of the first leagues of RoboCup was the two-dimensional soccer simulation league. In fact, two-dimensional soccer simulation league helped to address many different open problems of creating cooperative multi-agent systems. In RoboCup simulation league, teams of 11 autonomous software agents compete against each other by using RoboCup soccer server simulator software which is available from the official simulator website [2].
In the 2D environment, Nexus team focused on decision making and action selection module which is considered a high-level skill. The best action is the one that helps towards the agent's utmost success. The selected action has to bring about the most possible positive results in each simulation cycle, consistent with the definition of an ideal rational agent [3]. Every agent has to analyze various conditions as well as to handle newly received information. An intelligent agent should use the recently received information from the server in the best feasible way. It is possible that parts of the received information from the surrounding be of no use or of little importance. Considering parameters of each of the three possible actions (shooting, dribbling, and passing), the information received from the surrounding area and the existing conditions can be divided into two parts: The information that is related to only one *specific* action and the information that is *common* among all three actions [4].

2.1. One-phase decision making mechanism. In our one-phase evaluation method, we use a specific weight for each parameter that affects an action. Through test runs and analysis of the outcomes, we have experimentally obtained proper weights for these parameters. The analysis was aimed at pinpointing the weaknesses of our team and trying to adjust the weights to improve the efficiency of the system. Each weight can be either a reward or a punishment whose summation for each one of the possible actions can result in a computed priority that recommends the most reasonable action. To obtain the weights, we start with an initial value for each weight. Afterward, the agent is made to contest several times and after each contest, the weights are readjusted. This process is similar to the supervised learning [3], but it is performed offline. The weights will gradually adjust to a stable value. To evaluate the priority for each one of the possible actions, both specific and common measures are used. The highest calculated priority determines the preferred action.

2.2. Two-phase decision making mechanism. To determine the best action from amongst all possible ones for a given situation, we [4] first recognize the best of each action type, i.e., the best shoot, the best dribble, and the best pass, independently. It is clear that, when the best possible shoot is sought the parameters that affect the shooting action are considered, only. For dribble and pass actions a similar process is followed. In the next phase, we select the best of bests, i.e., the system chooses the best action from amongst the three best actions shoot, dribble, and pass. In this phase, common measures are used in order to evaluate the actions. Tab.1 shows the effects of different parameters on the three actions shoot, dribble, and pass. Fig.1 shows the overall work diagram.

[1] http://nexus.um.ac.ir/

Tab.1

Parameters' effects on different actions

Code	Parameter	Action
P1	Distance to the penalty point	Pass
P2	Receiver view angle	Pass
P3	Number of opponent around	Pass
P4	Adjacency rate to the goal	Pass
P5	Receiver attackness	Pass
P6	Pass distance	Pass
S1	Shoot speed	Shoot
S2	Attackness	Shoot
S3	Shoot distance	Shoot
S4	Shoot angle view	Shoot
D1	Number of opponent around	Dribble
D2	Distance to offside line	Dribble
D3	Agent stamina	Dribble
C1	Action interception probability	All
C2	Teammate density in target area	All
C3	Target area information novelty	All

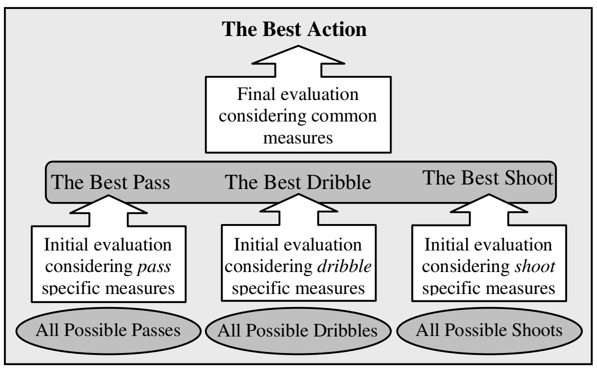

Fig.1 The two-phase selection diagram

2.3. Fuzzy two-phase decision making mechanism. We expected [5] the fuzzy system to be appropriate for decision-making process in the soccer simulation environment, considering the noise produced by the soccer server and uncertainties which affect all the perceptions and actions of the agents. Fuzzy systems are not sensitive to the completeness of the rule base, and even sometimes by removing half of the rules from a working system the performance does not considerably degrade, as long as the boundary rules are preserved in the fuzzy associative memory [6]. Our fuzzy rule base [5] includes 12 rules. The number of rules is much lower than the number of rules for our crisp system which is 50. For instance, the *high* priority measurement rules for the first phase are as the followings:

IF *P1 is Short* ***AND*** *P2 is High* ***AND*** *P3 is Low* ***AND*** *P4 is Long* ***AND*** *P5 is High* ***AND*** *P6 is Medium* ***AND*** *C1 is Low* ***AND*** *C2 is High* ***AND*** *C3 is High* ***THEN*** *Pass priority is High*

IF *S1 is Medium* ***AND*** *S2 is High* ***AND*** *S3 is Short* ***AND*** *S4 is High* ***AND*** *C1 is Low* ***AND*** *C2 is High* ***AND*** *C3 is High* ***THEN*** *Shoot priority is High*

IF *D1 is Low* ***AND*** *D2 is Short* ***AND*** *D3 is High* ***AND*** *C1 is Low* ***AND*** *C2 is High* ***AND*** *C3 is High* ***THEN*** *Dribble priority is High*

And the high priority measurement rule for the second phase as bellow:

IF *C1 is Low* ***AND*** *C2 is High* ***AND*** *C3 is High* ***THEN*** *selected action priority is High*

The proposed algorithm was implemented in our previous work [4]. Results of ten matches show that final scores of the team improved in the fuzzy approach. A team's success is directly influenced by each agent's actions. To calculate an agent's competence we should consider a measure that commensurates with the agent's pursuing goal [3]. To determine a team's efficiency, which in fact demonstrates the degree of the soccer agent's effectiveness, the game result or the two teams score difference can be the preferred approach. To compare the three mentioned methods, three teams were set up accordingly. To diminish the effect of accidental results, the fuzzy team was made to contest ten times with each non-fuzzy one.

Tab.2

The result of competitions between three different Nexus teams

Games	Ball possession for Nexus-3	Average within 10 matches
Nexus-1 vs. Nexus-3	69%	0.3 - 1.7
Nexus-2 vs. Nexus-3	57%	0.6 - 1.4
Nexus-1 : Nexus with one-phase decision making method Nexus-2 : Nexus with two-phase decision making method Nexus-3 : Nexus with fuzzy two-phase decision making method		

As Tab.2 shows, the results remarkably confirm the fuzzy method's superiority. In order to measure the accuracy of different actions 10 matches for each of the three Nexus teams played with three other teams. The result is shown in Fig.2 using the *"SoccerDoctor"* software [7] which is one of the best soccer simulation contest analyzers.

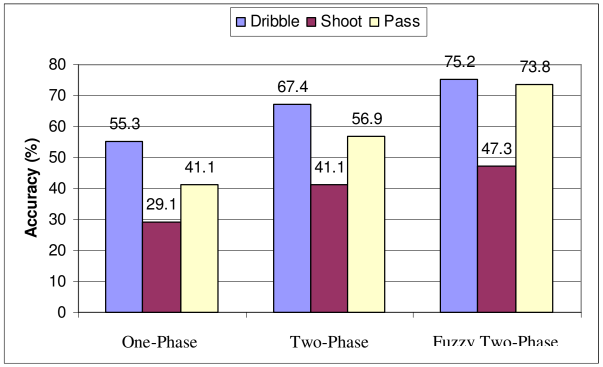

Fig.2 Average action accuracy within 10 matches

2.4. Game theory-based strategy making of coach agent. The most recent and final work done by Nexus on 2D simulation environment was developing a game theory-based data mining technique for strategy making of the soccer simulation coach agent [8]. A data mining process in the field of RoboCup soccer simulation involves gathering useful information out of the game data and acquires useful knowledge about the game situation known as strategy.

Game theory provides us with the mathematical tools to understand the possible strategies that utility-maximizing agents might use when making a choice. The simplest type of game considered in game theory is the *single-shot simultaneous-move* game. In this game all agents must take one action. All actions are effectively simultaneous. A single-shot game is a good model for the types of situations often faced by agents in a multi-agent system where the encounters mostly require coordination [9]. In a 2 player game, consider player *A* chooses a strategy and plays with it. Player *B* tries to learn *A*'s strategy and design his strategy as the best response to it. We assume *A* restricts itself to strategies realizable by *Deterministic Finite State Automata* (DFA). This is due to DFS strategies have been accepted widely as a model of bounded rationality [10, 11], and also learning the structure of an automaton has been shown to be a very hard problem [12].

In soccer simulation environment the coach agent is a privileged client used to provide assistance to the players [2]. There are two kinds of coaches, the *online coach* and the *trainer*. The trainer can exercise more control over the game and may be used only in the development stage, whereas the online coach connects to server during the game and provides additional advice and information to the players. The coach agent can control the play-mode, broadcast audio messages containing information, and getting noise-free information about the movable objects. The online coach is thus a good tool for opponent modeling, game analysis, and giving strategic tips to its team mates.

In our proposed model, the coach agent constructs a knowledge-base of the game in the main memory containing 11 game matrixes for each 11 soccer player agents and assumes opponent's strategy realizable by a DFA. The number of states in that DFA is a complexity measure. The coach would then apply the polynomial time learning algorithm of O(n) in which *n* is the number of states of the opponent automaton for all 11 game matrixes with respect to the payoffs assigned by the game knowledge-base as shown in Fig.3. A team strategy is mostly made using a knowledge-base or a set of <*state*, *action*> pairs. Using a special formation is another way in which each player has some predefined duties. These predefined duties are divided into static and dynamic.

	Intercept	Outplay	Pass	Shoot	Dribble	...
Intercept	1,1	0,3	0,4	0,4	0,3	...
Outplay	3,0	0,2	3,0	3,0	2,1	...
Pass	4,0	0,3	0,1	0,2	0,2	...
Shoot	4,0	0,3	0,1	0,1	0,2	...
Dribble	3,0	1,2	0,2	0,2	0,1	...
...	...	...	...	...	...	...

Fig.3 A sample agent game matrix

A team's success is directly influenced by each agent's actions. To determine team's efficiency, average results within 10 matches of three teams were set up accordingly. Nexus2005 [5] was build on Nexus2003 [4] with an improved fuzzy action section mechanism, and Nexus2006 takes advantage of a probabilistic action evaluation method. Tab.3 shows the results gained within 10 matches between each pair of the above teams.

Tab.3

The results of competitions within 10 matches

Games	Average within 10 games
StrategicNexus vs. Nexus2006	1.7 - 1.6
StrategicNexus vs. Nexus2005	1.9 - 1.6
StrategicNexus vs. Nexus2003	2.6 – 1.4

3. Applied Techniques to 3D Soccer Agents. Because of the simplified model of 2D simulation league, a three-dimensional physical simulation was created. The three-dimensional physical simulator used in Soccer Simulation League addresses additional classes of problems as well as global team behavior, decision making procedures and etc.
Based on the 2005 3D version work [13], Nexus team proposed a new scoring module [14] to select the best point on the goal line to shoot, considering player's position, catching and shooting time difference, and distance to target. To find the best point on the goal line to shoot, it is necessary to evaluate all points and obtain the one with the maximum calculated priority. Consequently, we designed an algorithm which firstly eliminates the points at which ball can not reach its target due to opponent interception.
As a rule of thumb, the shoot evaluation module deals with physical aspects of the ball controller agent, opponents, goalie, and the ball. Our aim is to find the best point on the goal line based on whose information if the ball is kicked, it will pass the goalkeeper ending inside the goal.
One of the parameters we need for the evaluation module is the temporal difference between ball and the goalie movement to reach the target. In other words, we calculate if the goalie reaches the target point sooner than the ball. This parameter would be then fed into the next fuzzy phase to estimate the catch probability. To do so, we subtract the time the agent takes to shoot considering rotation[2], from the time the goalie takes to reach the point and catch the ball. This subtraction trivially shows whether the ball will pass the goalie or will be intercepted. Let T_b be the time the ball takes to meet the target with the maximum speed, and T_r be the rotation time for the ball controller to adjust it's position beside the ball. T_g represents the time the goalie takes to catch the ball (Fig.4). Having calculated the above three parameters we define Δt as: $\Delta t = T_g - (T_b + T_r)$

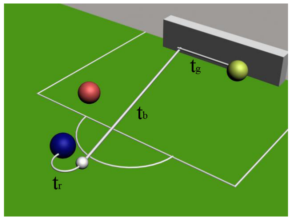

Fig.4 Temporal Measurements

[2] In 3D soccer simulation environment, unlike 2D version, agents are to be right behind the ball if they want to kick the ball straightly. In other words, agents can only kick the ball in the straight line which passes from the center of the ball and the center of player's body. By contrast, there is a kick direction in 2D system.

If $\Delta t > 0$, the ball would definitely pass the goalie and if $\Delta t < 0$, the ball would be intercepted. The greater the Δt, the higher probability of scoring goals. All these calculations were done assuming that there are no other agents except the goalkeeper in front of the ball controller to deviate the ball's direction.
In order to approximate the physical features of the environment, 100 of offline training test cases in which an agent shoots the ball from certain point toward the goal were done and results were saved on a log file. Having saved the above data, we try to formulate T_b, T_g, and T_r by means of interpolation. The Gaussian function $T_b(d)$ calculates the time takes the ball to pass distance d. Candidate shooting targets is a set of 25 points distributed along the goal line with 30cm interval. Fig.5 shows temporal difference measurement (Δt) through the goal line.

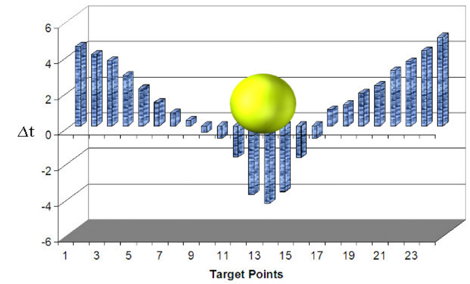

Fig.5 Temporal Difference Measurement (Δt) Through the Goal Line

In [8], we proposed a fuzzy approach to select the best shoot decision. It has shown that fuzzy systems provide a simple, efficient, and fast way of decision-making in comparison with the cumbersome and tedious process of applying many different rules for achieving the same results. We expected the fuzzy system to be appropriate for shoot evaluation process in the soccer simulation environment, considering the noise produced by the soccer server and uncertainties which affect all the perceptions and actions of the agents. Our fuzzy rule base includes 15 rules.

Tab.4

The result of 100 shoots

Number of Shoots	Simple Shoot Evaluation	Fuzzy Shoot Evaluation
10	6	7
20	11	13
30	13	18
40	19	23
50	22	28
Average Precision	42%	51%

To measure the shoot performance, *precision* measure was used as the ratio of the number of goal retrieved to the number of shoots through the goal expressed as a percentage. As Tab.4 shows, the results of 50 shoots comparing fuzzy approach and the non-fuzzy one, confirm the proposed method's superiority.

4. Applied Techniques to 3D Humanoid Agents. The current development of 3D Soccer Simulation League leads towards humanoid robots known as *soccerbot* agent, which already can be controlled by a lower level interface. However, controllers for these robots have to be developed in order to provide an easy-to-use interface. The rules has been maturated in many points and gained focus on the issues that are essential from a technical point of view. Thus, the center of mass of all robots has to be on a certain height in relation to the size of the feet. Fundamental for playing soccer are the abilities to walk and to kick. As body contact between the physical agents is unavoidable, the capability of getting up after a fall is also essential. For keeping a goal, the robot must be able to perform special motions.

4.1. Walking skill. Transferring the weight from one leg to the other, shortening the leg not needed for support, and leg motion along the walking direction are the key ingredients of this gait. Walking forward, to the side, and rotating on the spot are generated in a similar way. As the first step toward a skillful humanoid agent, walking is performed with a traditional control method that follows a set of generated ZMPs[3] along the path. This working dynamic model for biped robot walking is shown in Fig.6.

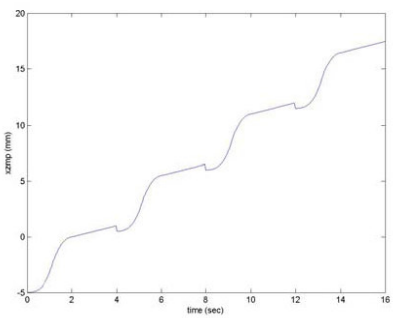

Fig.6 ZMP trajectory

The trajectory tracking methods (specially generated by a series of ZMPs) to control the agent balance while moving has been investigated in [15]. Generated trajectory is followed by a precise controller. The controller, knowing the exact path of the agent's joints, determines the velocity of the joint motors to direct different parts of the robot along the computed path. The walking skill of our agent is depicted in Fig.7.

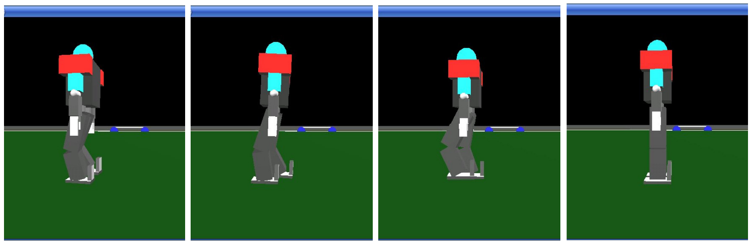

Fig.7 Soccerbot walking skill

4.2 Kicking skill. After inhibiting the walking behavior and stopping, the robot moves its weight to the non-kicking leg and then shortens the kicking leg, swings it back and accelerates forward. The kicking leg reaches its maximal speed when it comes to the front of the robot. Same principles for keeping robot's balance while walking or running are applied in performing actions like kick or dribble. The effectiveness of using dynamic methods like following the path generated by ZMPs with the help of new control methods like fuzzy PID control is already proved in such fields [16, 17].

4.3. Goalie dive skill. The goalie is capable of diving into both directions. First, it moves its center of mass and turns its upper body towards the left while shortening the legs. As soon as it tips over its left foot, it starts straightening its body again. While doing so it is sliding on its hands and elbows. These steps are depicted in Fig.8.

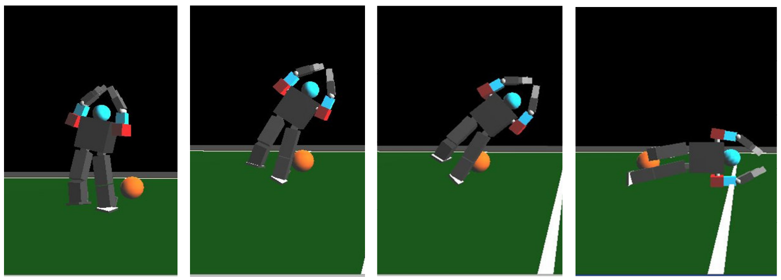

Fig.8 Soccerbot diving skill

[3] Zero Moment Point

5. Future work. Further improving the controller will be the next stage. Number of learning and optimizing methods such as artificial neural networks, genetic algorithms and other evolutionary approaches will be considered to give the controller an adaptive smooth behavior. For example genetic algorithm could be used to search the trajectory path, computed by the traditional dynamic model, with a small margin to achieve a better walking performance. Fuzzy logic, as a powerful tool in dealing with imprecise environments, can also improve the performance of the designed controller.

References: 1. *H. Kitano, Y. Kuniyoshi, I. Noda, M. Asada, H. Matsubara, and E. Osawa.* RoboCup: A challenge problem for AI. AI Magazine, 18(1):73–85, Spring 1997. **2.** *M. Chen, E. Foroughi.* Robocup Soccer Server manual 7.07, August, 2002. RoboCup Federation. available: http://sserver.sourceforge.net. **3.** *S. J. Russell, P. Norvig.* Artificial Intelligence A modern Approach, Prentice Hall, 1995. **4.** *V. Salmani, M. Naghibzadeh, F. Seifi, and A. Taherinia.* A Two-Phase Mechanism for Agent's Action Selection in Soccer Simulation, The Second World Enformatika Conference, WEC'05, Istanbul, Turkey, pp. 217-220, February 2005. **5.** *V. Salmani, A. Milani Fard, M. Naghibzadeh.* A Fuzzy Two-Phase Decision Making Approach for Simulated Soccer Agent, IEEE International Conference on Engineering of Intelligent Systems, pp. 134-139, Islamabad, Pakistan, April 22-23 2006. **6.** *B. Kosko*. Neural Networks and Fuzzy Systems, 1992, Englewood Cliffs: Prentice Hall. **7.** SoccerDoctor v1.0, developed by Wright Eagle, 2002. Available at: http://wrighteagle.org/sim/tools/SoccerDoctor.exe. **8.** *A. Milani Fard, V. Salmani, M. Naghibzadeh, S. K. Nejad, H. Ahmadi.* Game Theory-based Data Mining Technique for Strategy Making of a Soccer Simulation Coach Agent, the 6th International Conference on Information Systems Technology and its Applications, (ISTA2007), May 23-25, 2007, Kharkiv, Ukraine. **9.** *J. M. Vidal.* Learning in Multiagent Systems: An Introduction from a Game-Theoretic Perspective In Eduardo Alonso, editor, Adaptive Agents: LNAI 2636. Springer Verlag, 2003. **10.** *A. Rubinstein.* Finite automata play the repeated prisoner's dilemma. ST/ICERD Discussion Paper 85/109, London School of Economics, 1985. **11.** *A. Neyman.* Bounded complexity justifies cooperation in finitely repeated prisoner's dilemma. Economic Letters, pages 227-229, 1985. **12.** *M. J. Kearns and U. V. Vazirani.* An Introduction to Computational Learning Theory. MIT press, Cambridge, Massachusetts, 1994. **13.** *V. Salmani, F. Seifi, H. Moienzadeh, A. Milani Fard, M. Naghibzadeh.* Nexus 3D 2005 Team Description, RoboCup Soccer simulation team description, International Symposium of RoboCup, 2005, Osaka, Japan. **14.** *M. Mozafari, A. Milani Fard, V. Salmani, M. Naghibzadeh.* An Improved Fuzzy Mechanism for 3D Soccer Player Agent's Shoot Skill, IEEE INDICON 2006, September 15-17, 2006, New Delhi, India. **15.** *Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie.* Planning walking patterns for a biped robot, IEEE Tr. On Robotics and Automation, vol. 17, no. 3, pp. 280–289, 2001. **16.** *T. Yanase, T. Iba.* Evolutionary Motion Design for Humanoid Robots, GECCO'06, July 8–12, 2006, Seattle, Washington, USA. **17.** *A. Akramizadeh, A. Hosseini, M. Akbarzadeh.* Design, construction and fuzzy PID control of a 3DOF robot manipulator, CIS2001, Conference on Intell. Sys., Khaj-e-Nasir Univ., Tehran, IRAN, 2001.