A Fuzzy Two-Phase Decision Making Approach for Simulated Soccer Agent

Vahid Salmani Amin Milani Fard Mahmoud Naghibzadeh

Department of Computer Engineering, Software Simulation and Modeling Laboratory
Ferdowsi University, Mashhad, Iran

salmani@ferdowsi.um.ac.ir *milanifard@stu-mail.um.ac.ir* *naghib@ferdowsi.um.ac.ir*

Abstrat **- Soccer simulation is an effort to motivate researchers to perform artificial and robotic intelligence research; and at the same time put into practice and evaluate the results. The research includes design and implementation of robotic soccer simulation algorithms in a multi-agent environment. In this paper, we propose a fuzzy two-phase approach, based on our previous crisp two-phase mechanism, for the soccer player agent's action selection. To do so, a fuzzy decision making approach was implemented which tells an agent what action to take, in a given situation. In comparison with similar implementations, Our approach isfaster and simpler to follow. This method is integrated into our Nexus soccer simulation team of Ferdowsi University. The outcome showed the superiority of the fuzzy two-phase selection method compared to the non-fuzzy one.**

Keywords *- Multi-agent systems, fuzzy decision making, RoboCup soccersSimulation*

I. INTRODUCTION

Soccer simulation environment is a client-server platform which provides an excellent testbed to develop multi-agent systems. With this testbed, researchers need not get involved with the complexities of physical robot developmets. In RoboCup simulation league, many 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 [1].

For each player, there is a corresponding program which receives visual, audio, and other sensible information that are periodically sent by the server in every simulation cycle. The program has to analyze this information and perform whatever action it realizes to do [1]. The ball controller agent is capable of performing shoot, pass or dribble actions. Upon receiving new information from the server, the agent has to make a decision to perform one of these actions. It then informs the server of his decision in order for the server to update the playing environment as though the action has taken place.

Fuzzy systems have not been widly used in the annual RoboCup soccer simulation competitions for agents decision making process, yet. In this work, we proposed a new two-phase fuzzy approach based on our previous crisp two-phase mechanism [2]. We have 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.

II. DECISION MAKING PARAMETERS

There are two types of decision-making systems: individual and multi-person. In the latter case, which is also our matter of concern, an agent is not alone but it interacts with other agents whenever necessary [3].

The number of possible actions in each simulation cycle depends on many parameters such as information newness, action generation granularity, number of teammates, number of opponent players in the vicinity, overall team strategy, and coach instructions. Fig. 1 shows that the ball controlling agent has a variety of options for its next move. It has to analyze the situation and determine its best possible action and then communicate it with the server.

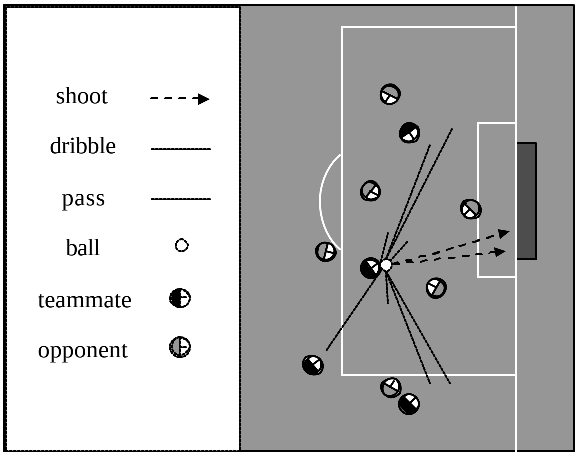

Fig.1. Feasible actions for the ball controlling agent

The best action is the one that helps towards the agent's utmost success. The attempt chosen has to bring about the most possible positive results in each simulation cycle, consistent with the definition of an ideal rational agent [4].

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 possible way. It is possible that parts of the received information from the surrounding be of no use or of little importance. For example, for the evaluation of the next possible actions the information that defines the area that is close to the ball controlling agent is more valuable than the information about far away distances. The *target area* refers to the region in which the ball keeps moving, while the action is in process.

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 [2]. The particular parameters of each action can be used as measures to evaluate different *feasible actions* and find out which one is the best. An action is considered to be feasible if, firstly it can be accomplished by the agent and, secondly, the ball can not be overtaken by the opponent during the action execution.

Common parameters can be used to evaluate and prioritize the three types of actions. Of the most important common parameters for the three actions are the density of the opponent players in that area, the probability of the ball interception by opponent players, and the newness of recently received information [5]. The density of the opponents in a region indicates the degree of the ability of action execution in that region. It is closely related to the probability that the opponent players being able to overtake the ball. Table 1 shows the effects of different parameters on the three actions shoot, dribble, and pass.

TABLE I
PARAMETERS EFFECT ONDIFFERENT 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

In this section, the evaluating measures for possible actions were studied. To be able to choose the best possible action, the soccer agent has to use an efficient real-time algorithm based on the mentioned measures.

III. RELATED WORKS

To evaluate possible actions, various methods have been suggested [2, 5, 6, and 7]. Given the specific measures as well as the common ones, we showed in [2] that there are two ways to evaluate each possible action: One-phase and two-phase decision-making mechanisms.

A. 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 ability 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. For example, in evaluation of the two actions A1 and A2, assuming A1 is better than A2, if evaluation module computes a higher priority for A2, the weights are adjusted by increasing the weights of those parameters which have more positive effect on A1 and decreasing those which have more positive effect on A2 (more negative effect on A1). This process is similar to the supervised learning [4], but it is performed offline. The weights will gradually adjust to a stable value.

Every parameter may have different values for different situations. Table 2 shows the weights of parameters with respect to the number of opponents in the target area and the length of the target area. These weights were experimentally obtained in our investigation based on many test runs.

TABLE II
WEIGHTS OF PARAMETERS WITH RESPECT TO THE OPONNET DENSITY AND LENGTH OF THE TARGET AREA

	5 units	10 units	More than 10 units
No player	0.15	0.19	0.25
1 player	0.10	0.14	0.19
2 players	-0.03	0.00	0.12
More than 2 players	-0.08	-0.03	0.05

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. As Fig. 2 shows, in this method, each feasible action's priority is computed as the summation of all related measures. The action with the highest priority is then recognized.

input: Array of feasible actions: passes, dribbles and shoots
output: Best action

```
max_priority ← 0
selected_action ← no_action
for each feasible action (FA) do
    priority ← 0
    for each evaluation measure (EM) do
        priority ←  priority + EM.weight
    end for
    if priority > max_priority then
        max_priority ← priority
        selected_action ← FA
    end if
end for
```

Fig. 2. One-phase action selection algorithm

These parameters may be adjusted so that the decision-making process follows a reasonable sequence of actions for limited number of situations. Since there are an unlimited number of different situations, it is not possible to adjust the weights so that the process works best all the times. On the other hand, affecting parameters varies for different actions.

In our experiments, we realized that if the decision-making process is broken into two phases, the number of parameters to deal with is reduced and the process is better managed. This lesson is what we learned by monitoring and analysis of numerous test runs. As the next section describes, the set of all affecting parameters are broken into two subsets, those that are common to all actions form the first one and others form the second one. It is worth mentioning that the second subset is different for different actions.

B. Two-phase decision making mechanism

To determine the best action from amongst all possible ones for a given situation, we first recognize the best of each action, 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.

input: Array of feasible actions: passes, dribbles and shoots
output: Best pass, dribble, and shoot

```
// First Phase
max_priority ← 0
selected_pass ← no_pass
for each Feasible Pass (FP) do
 priority ← 0
 for each Pass Specific Evaluation Measure(PSEM) do
    priority ←  priority + PSEM.weight
 end for
 if priority > max_priority then
    max_priority ← priority
    selected_pass ← FP
 end if
end for
```

* The above procedure would be applied to feasible dribbles and shoots as well, and selected pass, dribble and shoot information will be passed to the second phase

input: Selected pass, dribble and shoot actions
output: Best action

```
// Second Phase
pass_priority ← 0
dribble_priority ← 0
shoot_priority ← 0
selected_action ← no_action
for each Common Evaluation Measure (CEM) do
  pass_priority ← pass_priority + CEM.weight
  dribble_priority←dribble_priority + CEM.weight
  shoot_priority ← shoot_priority + CEM.weight
end for
if pass_priority > dribble_priority
and pass_priority > shoot_priority then
selected_action ← selected_pass
else if dribble_priority > shoot_priority then
  selected_action ← selected_dribble
else selected_action ← selected_shoot
end if
```

Fig. 3. Two-phase action selection algorithm

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 actions.

Fig. 3 shows the two-step evaluation method in which in the first phase it finds the best possible shoot, pass and dribble using specific measures. In the second phase, it selects the actual action to take, using common measures. To determine the priority in the second step, the calculated priorities in the first step is not considered.

Method (A) consumes a lot more processing time than method (B). Therefore, it does not leave any time for the simulator to augment further precision and to increase intelligence. If the number of alternatives to be compared becomes large, the soccer agent might not be able to complete the evaluation process in one simulation cycle period. Note that the two mentioned methods were explained without considering the overall team's strategy and the coach's guidance. To evaluate the actions, while considering team's strategy and the coach's guidance, other parameters have to be added to the list of parameters affecting the evaluation process.

IV. Fuzzy Decision-Making Approach

Fuzzy sets were first introduced by Zadeh as a means of representing and manipulating data that is not precise, but rather fuzzy [8]. We expected 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.

To solve this problem, we have used some concepts and techniques from *fuzzy logic theory* [9,10] as opposed to crisp set theory. The following steps is taken in order to design any fuzzy system [10,11].

A. Membership fuctions and linguistic variables

Definition of membership functions and linguistic variables are the first steps in fuzzy system designing. Each linguistic variable contains terms which are interpretation of technical figures. It is important to choose a suitable membership function which best maps each value of the technical figure to a membership degree of the corresponding linguistic term. In our work, we have used different types of membership fuctiones for different fuzzy terms which are available in appendix A at the end of the paper.

B. Fuzzy rule base

The second step in designing a fuzzy system is the creation of a fuzzy logic rule base which supplies the knowledge of the system. 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 degrade, as long as the boundary rules are preserved in the fuzzy associative memory [12].

To build the rule base, we need to review the standard methods. A fuzzy logic rule is an if-then rule. The *if* part is a fuzzy predicate which defined in terms of linguistic values and fuzzy operators *Intersection* (*t-norm*) and *Union* (*s-norm*). The *then* part is called the *consequent*. There are many implementations of fuzzy union and intersection operators. In this work we have used product t-norm [13] as our aggregate method which is described as follows:

$$t-norm = \prod_{i=1}^{n} m_i \quad (1)$$

Our fuzzy rule base includes 12 rules. The number of rules is much lower than the number of rules for our crisp system which was 50. Nine of these fuzzy rules are considered in the first phase of our method and the three remaining rules are used in the second phase. 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 is:

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

C. Defuzzification method

The third step in the design of a fuzzy system is choosing an appropriate defuzzification method [3,14]. The objective of a defuzzification method is to derive the non-fuzzy (crisp) value that best represents the fuzzy value of the linguistic output variable. Center of area (CoA), center of maximum (CoM) and mean of maximum (MoM) are some available defuzzification methods. In our work, we have decided to employ CoM for the decision making approach.

$$W = \frac{\sum_{i=1}^{n} m_i W_i}{\sum_{i=1}^{n} m_i} \qquad (2)$$

D. Algorithm design

Fig. 4 shows our proposed algorithm which consists of two major phases.

V. SIMULATION AND RESULTS

The proposed algorithm was implemented in C++ on *Nexus soccer simulation team* [2]. Experiments were done under Linux RedHat 9 distribution operating system on a desktop computer with Pentium 4 CPU 2.5GHz and 1GB RAM. Results of ten games show that final scores of the team improved in the fuzzy approach. An important point to mention is that, when it comes down to evaluate the performance of a system like this soccer simulation, the final score of the match is not always a good measuring paprmeter. Many factors affect the score. 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 [4].

Each soccer agent attempts to change the game towards its own advantage so that it not only minimizes losing but also maximizes scoring through imposing maximum pressure on the opponent's goal. In other words, the soccer agent's entire endeavor is winning the game. Therefore, 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. As table 3 shows, the results remarkably confirm the fuzzy method's superiority.

input: Array of feasible actions: passes, dribbles and shoots
output: Best pass, dribble, and shoot

```
// First Phase
max_priority ← 0
selected_pass ← no_pass
for each Feasible Pass (FP) do
 priority ← 1
 for each Pass Specific Evaluation Measure(PSEM)do
    priority ←  priority * m PSEM
 end for
 W1 ←  m high_priority
 W2 ← m medium_priority
 W3 ← m low_priority
 W  ← (W1*high_priority + W2*medium_priority +
        W3*low_priority) /
 (high_priority + high_priority + High_priority)
 if  W > max_priority then
    max_priority ← W
    selected_pass ← FP
 end if
end for
```

* The above procedure would be applied to feasible dribbles and shoots as well, and selected pass, dribble and shoot information will be passed to the second phase

input: Selected pass, dribble and shoot actions
output: Best action

```
// Second Phase
pass_priority ← 0
dribble_priority ← 0
shoot_priority ← 0
selected_action ← no_action
for each Common Evaluation Measure (CEM) do
   pass_priority ← pass_priority * m CEM
   dribble_priority ← dribble_priority * m CEM
   shoot_priority ← shoot_priority * m CEM
end for
for each action do
  W1 ←  m high_priority
  W2 ← m medium_priority
  W3 ← m low_priority
  W_action←(W1*high_priority + W2*medium_priority +
           W3*low_priority) /
   (high_priority + high_priority + high_priority)
  if  action is pass then pass_priority ← W_action
  else if  action is dribble then
     dribble_priority ← W_action
  else  shoot_priority ← W_action
  end if
end for
if pass_priority > dribble_priority
     and pass_priority > shoot_priority then
selected_action ← selected_pass
else if dribble_priority > shoot_priority then
    selected_action ← selected_dribble
else selected_action ← selected_shoot
end if
end for
```

Fig. 4. Fuzzy Two-phase action selection algorithm

TABLE III
THE RESULT OF COMPETITION BETWEEN THREE NEXUS TEAMS

Games	Ball possession for Nexus-3	Average within 10 games
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		

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. 5 using the "*SoccerDoctor*" software [15] which is one of the best soccer simulation contest analyzers.

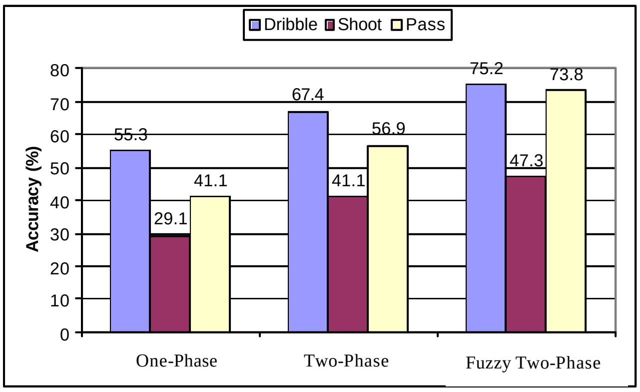

Fig. 5. Average action accuracy within 10 matches

It is notable that the fuzzy logic membership computations do not add to the complexity in the asymptotic sense. Therefore, the fuzzy two-phase decision making process has the same complexity as the crisp one, which has a worst case value of $O(n.m)$ where n is the number of feasible actions and m represents the number of parameters. However, since number of fuzzy approach rules is about one fifth of the non-fuzzy one, this leads to the run time and development process speed up. Fig. 6 shows T_f / T_{nf} which is rate of decision module average run time during 10 matches in where T_f is fuzzy decision module average run time and T_{nf} is non-fuzzy decision module average run time in the same match. Results as expected shows this rate is always less than 1 which means the fuzzy approach has a less overhead in contrast with the non-fuzzy method.

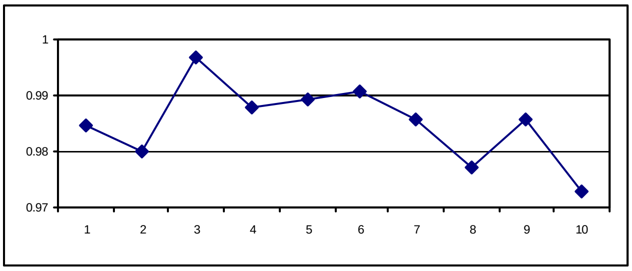

Fig. 6. Rate of decision module average run time during 10

VI. CONCLUSION

Common measures are essential but not sufficient to evaluate various soccer player's action. To evaluate these actions more precise and specific measures are needed. In this research, we concentrated on three action types shoot, dribble, and pass. We have devised a fuzzy two-phase action selection method based on our previous experiments and compared its efficiency with the crisp one-phase and two-phase evaluation methods. Three Nexus teams were set up and competed many times and the results were summarized. The outcome clearly showed the superiority of the fuzzy two-phase selection method.

ACKNOWLEDGMENT

Authors would like to thank members of *Nexus*, the Ferdowsi University RoboCup soccer simulation team, for their modest cooperation and special thanks to *Young Iranian Elites Association* NGO for supporting young Iranian students' scientific works and projects.

REFRENCES

[1] M. Chen, E. Foroughi,, Robocup Soccer Server manual 7.07, August, 2002. RoboCup Federation. available: http://sserver.sourceforge.net.
[2] 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.
[3] H.-J. Zimmermann, "Fuzzy Sets, Decision Making, and Expert Systems," 1987.
[4] S. J. Russell, P. Norvig, *Artificial Intelligence A modern Approach*, Prentice Hall, 1995.
[5] L. P. Reis, N. Lau, FC Portugal Team Description: RoboCup 2000 Simulation League Champion, FC Portugal 2000 Team Description Paper. available: http://www.ieeta.pt/robocup/documents/FCPortugalChampion.ps.zip
[6] P. Stone, D. McAllester, "An Architecture for Action Selection in Robotic Soccer," pp. 1-8. AT&T Labs - Research. Florham Park, NJ, 2001.
[7] P. Stone, Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer, PhD Thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, 1998. Available as technical report. Available: http://www.cs.utexas.edu/~pstone/book/
[8] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," IEEE Transaction Systems Man. Cybern, vol SMC-3, no. 1, pp. 28-44, 1973.
[9] L. A. Zadeh, "Fuzzy Sets", Information and Control, 1965. 338-353
[10] L. Wang, A Course in Fuzzy Systems and Control, Prentice-Hall, 1997
[11] C. V. Altrock, "Fuzzy Logic & Neurofuzzy Applications Explained," 1995.
[12] B. Kosko, Neural Networks and Fuzzy Systems, 1992, Englewood Cliffs: Prentice Hall
[13] J.-S. R Jang, C.-T Sun, E. Mizutani, "*Nuro-Fuzzy and Soft Computing*". Englewood Cliffs, NJ: Prentice-Hall, 1997.
[14] R.M Tong, and P.P.Bonissone, "Linguistic solutions to Fuzzy Decision Problem," in Zimmermann, Zadeh and Gines (Ed.), "Fuzzy sets and Decision Analysis," 1984.
[15] SoccerDoctor v1.0, developed by Wright Eagle, 2002. Available at: http://wrighteagle.org/sim/tools/SoccerDoctor.exe

APPENDIX A

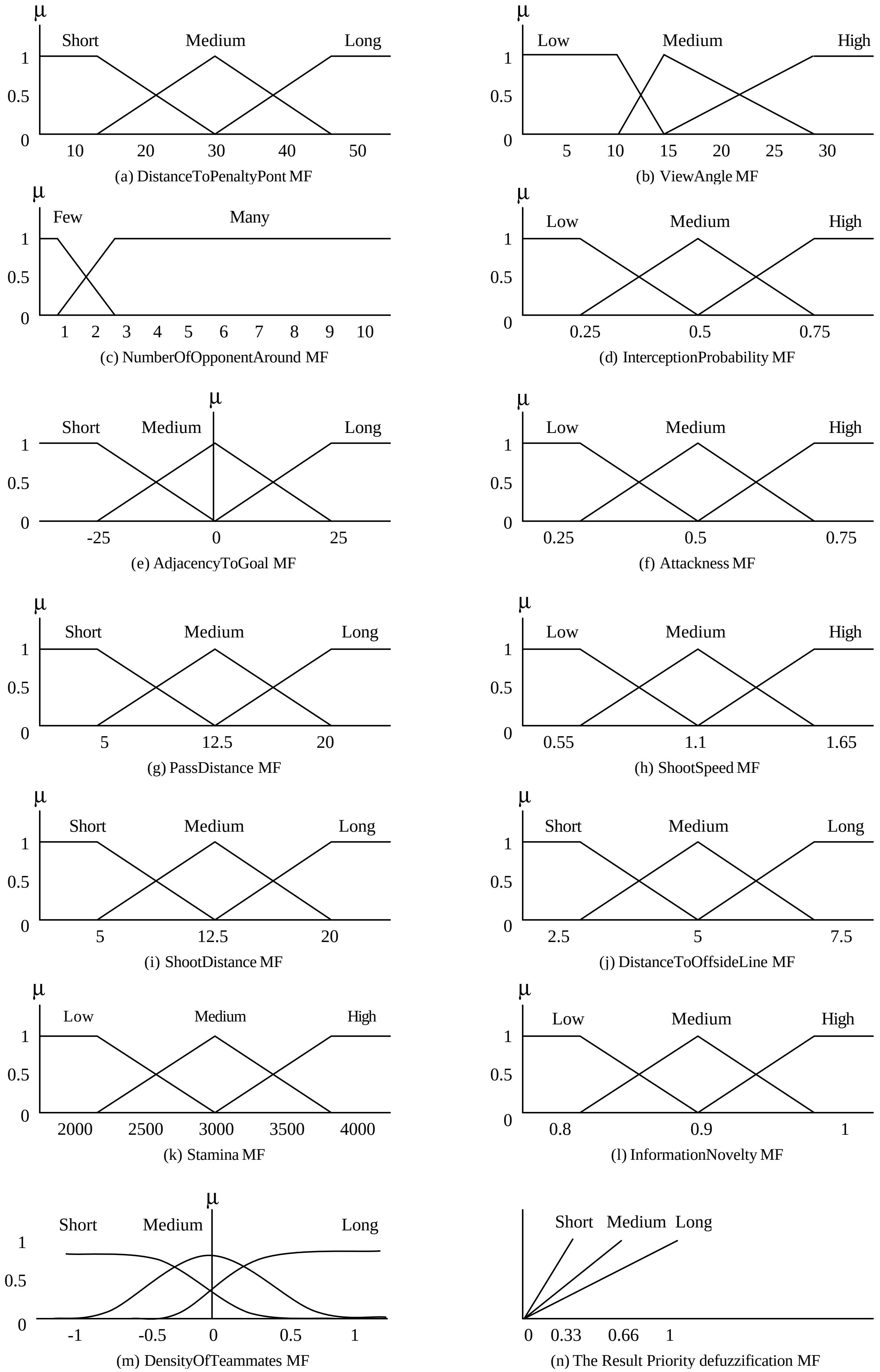

Fig. 7. Experimental membership functions we used in Nexus 3