

# Fuzzy Inference Based Forecasting in Soccer Simulation 2D, the RoboCup 2015 Soccer Simulation 2D League Champion Team

Xiao Li<sup>(✉)</sup> and Xiaoping Chen

Department of Computer Science,  
University of Science and Technology of China, Hefei, China  
runtu@mail.ustc.edu.cn, xpchen@ustc.edu.cn

**Abstract.** WrightEagle is the champion of RoboCup 2D simulation league, the latest technic used in WrightEagle will be presented in this paper. This paper introduces the application of fuzzy inference system in WrightEagle. The fuzzy inference system is built without knowledge of the mathematical description of the opponents. With the fuzzy inference system, we are able to forecast the actions of the opponents in some aspects. Once we can forecast the actions of the opponents, we can make corresponding decisions.

## 1 Introduction

In RoboCup 2D simulation league, every team consists of 12 autonomous agents (11 players and an online coach). For each agent, its goal is to make the best decision with the information observed. Up to now, Markov Decision Processes framework [1], multi-step decision-making framework [2], MAXQ hierarchical structure [3] and heuristic online planning [4,5] techniques have been applied on WrightEagle to make decisions. To make better decisions, we want to forecast the actions or decisions made by the opponents. Apparently, as in a competition, if the agent is able to forecast the actions or decisions made by the opponents, it will be very helpful for the agents to make corresponding decisions. To this end, an intuitive method is to build a mathematical model for the opponent. But the scale of 2D simulation is so large that it is nearly impossible to build a mathematical model for the opponent completely. Then we come up with the fuzzy inference system. The fuzzy inference system makes it possible to map inputs (observations) to outputs (actions) without knowledge of the mathematical description of the opponent. With the rules or knowledge we concluded in the past, we can build some fuzzy inference systems. These systems can help us to forecast opponents' actions in some aspects.

## 2 Fuzzy Inference System (FIS)

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic [6]. In fuzzy logic, a statement is able to assume

any real value between 0 and 1, representing the degree to which an element belongs to a set. So it is able to work with human inputs. On the other hand, fuzzy logic does not need complicated mathematical models. So the FIS is easy to implement.

Mamdani-Type Fuzzy Inference and Sugeno-Type Fuzzy Inference are two of the most common types of fuzzy methodology. In the following sections, we will use these two methods to deal with a simple and specific scene in the 2D simulation.

## 2.1 Mamdani-Type Fuzzy Inference

The Mamdani's fuzzy inference method is the most common fuzzy methodology. It was first proposed by Ebrahim Mamdani [7] in 1975. A Mamdani's fuzzy inference system is generally composed of five steps:

1. Determine a set of fuzzy rules,
2. Fuzzify inputs with the input membership functions,
3. Obtain each rule's conclusion,
4. Aggregate conclusions obtained in step 3,
5. Defuzzification.

The detailed description of these steps will be introduced in the following part.

**Determine a Set of Fuzzy Rules.** The form of fuzzy rules in Mamdani's fuzzy inference system is always like this:

*If  $x$  is  $A$  and/or  $y$  is  $B$ , then  $z$  is  $C$ .*

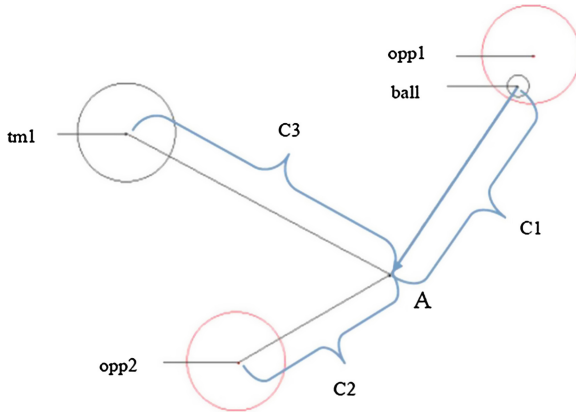
where  $A$ ,  $B$  and  $C$  are linguistic values defined in the fuzzy set.

For such a scene in the 2D simulation (Fig. 1), we assume that opponent1 (opp1) is attending to pass the ball to opponent2 (opp2) to point A. Our teammate1 (tm1) is marking on opponent2. The goal of tm1 is to judge if opp1 would pass the ball to point A. Here, we assume the possibility of opp1 to pass the ball successfully to be the evaluation of opp1's action.

It is intuitive that the possibility of opp1 to pass the ball to opp2 successfully (P) is positively related to  $\Delta c1 = c1 - c2$  and negatively related to  $\Delta c2 = c1 - c3$ . So two simple rules can be concluded here:

*rule1: If  $\Delta c1$  is large and  $\Delta c2$  is small, then  $P$  is high.*

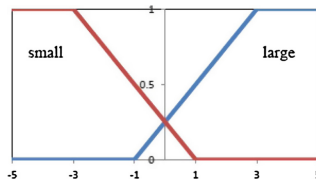
*rule2: If  $\Delta c1$  is small or  $\Delta c2$  is large, then  $P$  is low.*



**Fig. 1.** A specific scene in 2D simulation. c1 is the cycle needed by the ball to get to point A; c2 is the cycle needed by opp2 to get to point A; c3 is the cycle needed by tm1 to get to point A.

**Fuzzify Inputs with the Input Membership Functions.** For a fuzzy set  $A$ , its membership function is defined as  $\mu_A : X \rightarrow [0, 1]$ . Set  $X$  is the universe of discourse. For each  $x \in X$ , the value  $\mu_A(x)$  is called the degree of membership of  $x$  to fuzzy set  $A$ . For example, if  $\mu_A(x) = 0.7$ , we can say that  $x$  belongs to  $A$  to a degree of 0.7.

With the fuzzy set, we can describe fuzzy concept like “large” and “small”. Therefore, the FIS is able to deal with human inputs. For the “large” and “small” in the rules, their membership functions are defined in Fig. 2.



**Fig. 2.** The membership functions of “small” and “large”.

Using the fuzzy sets defined in Fig. 2, we can fuzzify the inputs. For example, with the inputs  $\Delta c1 = 1$ ,  $\Delta c2 = -2$ , we can obtain the membership values for each rule (Fig. 3).

The fuzzy operators “and” (min) and “or” (max) are used in Fig. 3, then we obtain a single membership value for each rule.

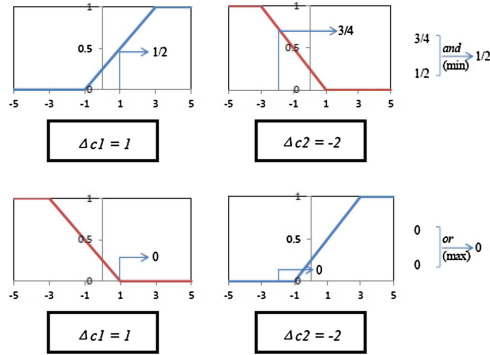


Fig. 3. Obtain a single membership value for each rule.

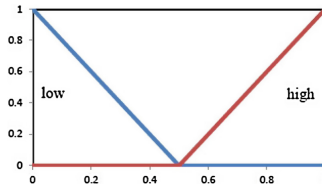


Fig. 4. The membership functions of “low” and “high”.

**Obtain Each Rule’s Conclusion.** To obtain each rule’s conclusion, we should define the membership functions of the outputs first. For the “high” and “low” in the rules, their membership functions are defined in Fig. 4.

Then, with these membership values obtained in step 2, we can use a fuzzy implication operator (min) to obtain a fuzzy set for each rule (Fig. 5).

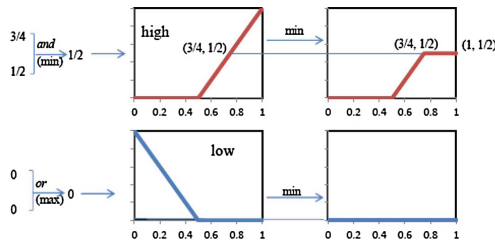
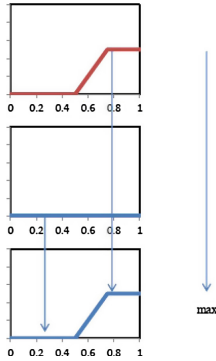


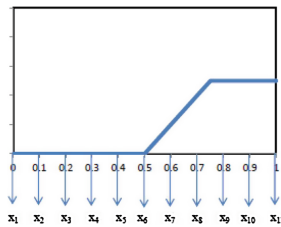
Fig. 5. Obtain a fuzzy set for each rule.

**Aggregate Conclusions.** We choose the max to be the aggregation operator, then we can combine the results obtained in last step (Fig. 6).



**Fig. 6.** Aggregate conclusions.

**Defuzzification.** The result of last step is a fuzzy set, it is not we want when we need to make a decision. As in the scene mentioned above, we do not want to know the P is high or low but the accurate value of P. Therefore, we need to transform the fuzzy set into an accurate value. The process of this transformation is called defuzzification. Here, we use the centroid method, which is one of the most common used methods for defuzzification. Centroid method is to compute the center of the mass. Here, we take some sample points to calculate the center (Fig. 7).



**Fig. 7.** Aggregate conclusions.

$$P = \frac{\sum_{i=1}^{11} x_i \times \mu(x_i)}{\sum_{i=1}^{11} \mu(x_i)} = 0.83 \tag{1}$$

Now, the accurate value of P is obtained. So we can say that the possibility of opp1 to pass the ball to opp2 successfully is 0.83. Then, we can forecast that the possibility of opp1 to pass the ball to point A is 0.83. In other words, now we are able to forecast the possibility of an opponent to pass the ball to a certain

point. Therefore, we can forecast to which point an opponent will pass the ball through enumeration.

### 2.2 Sugeno-Type Fuzzy Inference

The Sugeno fuzzy inference method was first proposed in 1985 [8], it is very similar to the Mamdani’s method. Generally, the Sugeno fuzzy inference system is composed of four steps:

1. Determine a set of fuzzy rules,
2. Fuzzify inputs with the input membership functions,
3. Obtain each rule’s conclusion,
4. Aggregate conclusions obtained in step 3.

**Determine a Set of Fuzzy Rules.** The form of the rules in Sugeno method is like this:

$$\text{If } x \text{ is } A \text{ and/or } y \text{ is } B, \text{ then } z \text{ is } f(x, y).$$

Unlike the Mamdani’s method, the output of Sugeno method is no longer a fuzzy set. It is a function of the inputs. So there is no need of defuzzification. For the same scene mentioned above, we can conclude two rules:

$$\text{rule1: If } \Delta c1 \text{ is large and } \Delta c2 \text{ is small, then } P = f(\Delta c1 - \Delta c2).$$

$$\text{rule2: If } \Delta c1 \text{ is small or } \Delta c2 \text{ is large, then } P = g(\Delta c1 - \Delta c2).$$

where  $f(x)$  and  $g(x)$  are defined in Fig. 8.

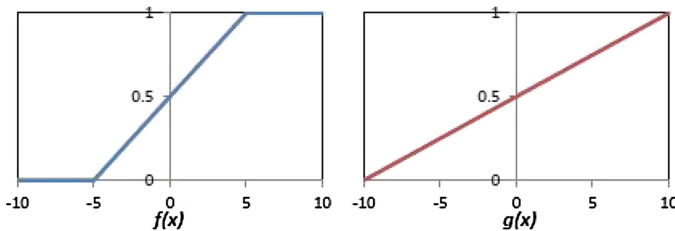


Fig. 8.  $f(x)$  and  $g(x)$ .

**Fuzzify Inputs with the Input Membership Functions.** In this step, the Sugeno method is exactly the same with Mamdani’s method (Fig. 9).

**Obtain Each Rule’s Conclusion.** Since the output is a function of the inputs, it is much simpler to obtain each rule’s conclusion.

$$\text{rule1 : } P = f(\Delta c1 - \Delta c2) = 0.8$$

$$\text{rule2 : } P = g(\Delta c1 - \Delta c2) = 0.65$$

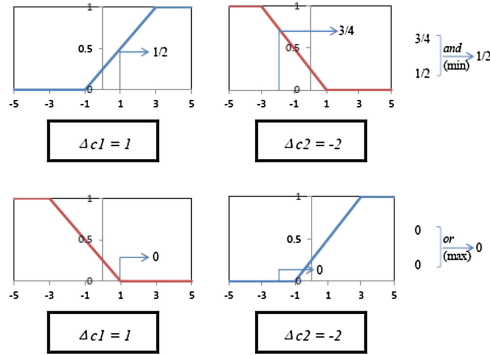


Fig. 9. Obtain a single membership value for each rule.

**Aggregate Conclusions.** In Sugeno method, aggregation is to compute the weighted average of the outputs obtained is last step. The membership value obtained in step 2 is used to be the weight of each rule ( $w_1 = 1/2, w_2 = 0$ ).

$$P = \frac{\sum_{i=1}^2 w_i \times p_i}{\sum_{i=1}^2 w_i} = 0.8 \tag{2}$$

### 2.3 Comparison Between Mamdani and Sugeno

Based on the form of the rules, we can find out that the Mamdani’s method is more intuitive. But in 2D simulation, every decision should be made within 100 ms. If the number of the rules is large, it will take much time to compute. So we should apply the Mamdani’s method on scenes with small number of rules. Science the output is a function of the inputs in Sugeno method, it is computationally efficient. So it is able to handle with scenes with large number of rules.

## 3 Experiment

We have implemented a FIS and applied it on the marking module of WrightEagle. To test the FIS, we played 2000 full games (1000 before we implemented the FIS and 1000 after) with each team (Helios2014, Yushan2014, Gliders and Oxsy). The binary code of Helios2014, Yushan2014, Gliders and Oxsy we used were released by RoboCup 2014. These games were played on the same hardware and software environment. For each team, we take account average goal and win rate. The results are shown in Tables 1 and 2.

Table 1 shows that, after we implemented the FIS, the win rate against opponent teams increased. The increase of win rate shows that the FIS works well.

What's more, Table 2 shows that the average goal of opponent teams decreased, from which we can conclude that the FIS did increase the defensive ability of WrightEagle.

**Table 1.** Win rate against each team

	Before	After
Helios2013	58.3 %	66.6 %
YuShan2013	84.8 %	88.2 %
Gliders	71.4 %	82.2 %
Oxsy	90.3 %	94.9 %

**Table 2.** Average goal against each team

	Before	After
Helios2013	2.38:1.47	2.41:1.15
YuShan2013	3.30:0.77	3.43:0.64
Gliders	2.34:0.77	2.57:0.74
Oxsy	4.64:0.81	4.63:0.79

## 4 Related Work

As for a FIS, how to get rules is the most important challenge. Soccer is the most popular sport in the world. The sport has developed for hundreds of years. There are so many matches from which we can conclude rules. The human players and coaches have come up with numbers of strategies. So we can conclude some general rules from these strategies. On the other hand, the 2D simulation has developed for decades, we can conclude rules from different teams and different matches. The machine learning techniques and big data techniques also can be used here.

The FIS we talked above is a multi-input, single-output model. There are some scenes in 2D simulation in which we need use multi-input, multi-output models. Actually, a multi-input, multi-out model can be separated into some multi-input, single-output models.



## 5 Conclusion

For the purpose of forecasting actions of opponents in soccer simulation 2D, we introduced Mamdani-Type Fuzzy Inference and Sugeno-Type Fuzzy Inference to deal with a specific scene in soccer simulation 2D. Then we implemented and applied a FIS to the marking module of WrightEagle. Based on the result of the experiment, we conclude that the FIS improves the defensive ability of our team. It did work like what we expected. The most important advantage of FIS is that it can work well without the knowledge of the mathematical description of the opponent. Another advantage of the FIS is that, when we want to take account more things, what we need to do is to add new rules or edit old rules. Then we do not have to take much time to write a new algorithm.

## References

1. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming, vol. 414. Wiley, Hoboken (2009)
2. Zhang, H., Chen, X.: The decision-making framework of WrightEagle, the RoboCup 2013 soccer simulation 2D league champion team. In: Behnke, S., Veloso, M., Visser, A., Xiong, R. (eds.) RoboCup 2013. LNCS, vol. 8371, pp. 114–124. Springer, Heidelberg (2014)
3. Dietterich, T.G.: The maxq method for hierarchical reinforcement learning. In: ICML, pp. 118–126. Citeseer (1998)
4. Bai, A., Wu, F., Chen, X.: Online planning for large mdps with maxq decomposition. In: Proceedings of the Autonomous Robots and Multirobot Systems Workshop, at AAMAS 2012, June 2012
5. Bai, A., Wu, F., Chen, X.: Towards a principled solution to simulated robot soccer. In: Chen, X., Stone, P., Sucar, L.E., van der Zant, T. (eds.) RoboCup 2012. LNCS(LNAI), vol. 7500, pp. 141–153. Springer, Heidelberg (2013)
6. Zadeh, L.A.: Fuzzy sets. *Inf. Control* **8**(3), 338–353 (1965)
7. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Mach. Stud.* **7**(1), 1–13 (1975)
8. Sugeno, M.: *Industrial Applications of Fuzzy Control*. Elsevier Science Pub. Co., Amsterdam (1985)