ROBOT BEHAVIOR COORDINATION BASED ON FUZZY DECISION-MAKING

Nelson Ramos[1] Pedro U. Lima[1] João M.C. Sousa[2]

[1] *Instituto de Sistemas e Robótica*
[2] *Dept. of Mechanical Engineering, GCAR/IDMEC*
Instituto Superior Técnico, Technical University of Lisbon
Av. Rovisco Pais, 1049-001 Lisbon, PORTUGAL

Abstract: This paper introduces a fuzzy decision-making algorithm for robot behavior coordination. The algorithm belongs to the arbitration class of behavior coordination mechanisms, under which only one behavior is running at a time. However, it is possible to use a hierarchical decision mechanism for hierarchical behaviors without interference between hierarchical levels. With this fuzzy decision method it is possible to represent a specific model of the world where the robot evolves. This algorithm consists of defining a set of behaviors, a set of world states, a cost function for behaviors, a set of goals, and a set of constraints. For each behavior and actual world state pair, a cost function is computed. The cost of each pair is evaluated by the overall goals. Goals and constraints are aggregated using a fuzzy operator and the optimal choice is the behavior with the maximum resulting value. This algorithm was tested with success in realistic simulations of a goalkeeper soccer robot.

1. INTRODUCTION

Fuzzy logic has been applied in behavior-based mobile robot control (Vadakkepat *et al.* 2004, Pirjanian 1999). A behavior is most commonly described as a set of purposive perception-action pars. The main advantages of fuzzy logic is its capability to handle vague descriptions producing controllers not prone to conventional design methods. Furthermore, fuzzy logic allows to define behavior decision rules through linguistic terms that simplify expert knowledge encoding. Fuzzy decision in behavior coordination is usually classified as a command fusion coordination mechanism, see (Pirjanian 1999, Pirjanian and Mataric 1999), but it could also be used as an arbitration fusion coordination mechanism. By using a command fusion approach, problematic situations could arise; when a fuzzy decision is made with command fusion of two competitive behaviors, as in (Pirjanian and Mataric 1999), the results can be inconsistent, because the final action is a new behavior that is not described. For example, in the soccer domain, when it is possible to execute a clear ball and execute also an outlet pass, if command fusion is made, the result can be a ball kicked towards an intermediate direction which is the average of both directions. If an opponent is in this direction, the robot will pass the ball to it which is the worst decision. Various methods have been proposed to solve this problem in (Pirjanian 1999), but none is intuitive enough. Some architectures apply distributed hierarchical fuzzy inference with command fusion, instead of centralized fuzzy decision, as in (Vadakkepat *et al.* 2004). This approach has the advantage of abstraction, but does not guarantee sequencing of primitive actions corresponding to the same behavior.

This paper proposes a generic mechanism that allows distributed hierarchical fuzzy inference and guarantees sequencing of primitive actions of the same behavior. It guarantees also that only one behavior is selected and only described behaviors are executed, instead of a combination of behaviors. Furthermore, it enables modeling and considers general constraints at run time.

The paper is organized as follows. In Section 2, our behavior coordination method based on fuzzy decision-making is introduced. This section explains relations between behaviors, world states and cost function. Goals are constraints are described, as well as how fuzzy decision making is applied in this work. An extension to a hierarchical environment is discussed. Section 3 introduces a case study on a robotic goalkeeper. Simulation results are presented in Section 4 and Section 5 draws some conclusions.

2. BEHAVIOR COORDINATION BASED ON FUZZY DECISION-MAKING

Fuzzy decision making deals with non-probabilistic uncertainty and vagueness in the environment in which the decision making takes place. Two important elements of decision making are the goals of the decision that are represented by the maximized objective function and the imposed constraints that confine the search space. Fuzzy decision making essentially replaces the crisp goals and the constraints with their fuzzy equivalents (Bellman and Zadeh 1970, Sousa and Kaymak 2002).

This paper using the fuzzy decision making framework to obtain an optimal decision in coordinate behavior. The optimal behavior will be the one that better satisfies the goals and constraints of the system. In this paper, fuzzy inference only selects one behavior at each time step. Thus, this mechanism belongs to the arbitration class of behavior coordination mechanisms (Pirjanian and Mataric 1999).

2.1 Behaviors, World States and Cost Function

A behavior, b, and a world state, $\mathbf{x}$, are mapped in a cost function, f, such that

$$f : (b, \mathbf{x}) \rightarrow]-\infty, +\infty[. \tag{1}$$

This means that for each pair $(b, \mathbf{x})$ the function f has a value representative of behavior b cost when the world is in state $\mathbf{x}$. The state $\mathbf{x}$ is a multidimensional value, which means that $\mathbf{x}$ represents all world component states that are relevant for the defined behaviors.

The cost function is multidimensional, so it is important to guarantee that it has some properties, such as dimension independence of behaviors and states. This independance can be defined as follows: when the dimension of behaviors is p, the dimension of states is q, and p increases to $p + 1$, the cost and definition of old behaviors are not affected. A similar property holds for the states: when more states are added, the cost and definition of old behaviors are not affected. These properties are very important to assure the expansibility of behaviors.

Therefore, a cost function can be defined as

$$f(b_i, \mathbf{x}) = \sum_{j=1}^{n} r_{i,j}(\mathbf{x}) \tag{2}$$

where $r_{i,j}(\mathbf{x})$ is the cost of rule j for behavior b_i. The cost $r_{i,j}(\mathbf{x})$ can be described as

$$r_{i,j}(\mathbf{x}) = k_{i,j} d_{i,j}(\mathbf{x}), \tag{3}$$

where $k_{i,j}$ is a constant behavior b_i is the cost factor for rule j, and $d_{i,j}(\mathbf{x})$ is an activation level for rule j. The activation level of a rule, $d_{i,j}(\mathbf{x})$ can be described as a predicate that operates in fuzzy states. So, we consider $\mathbf{x}$ to be a vector of fuzzy states with dimension m. Each fuzzy state is represented by a linguistic term. So, the predicate that describes $d_{i,j}(\mathbf{x})$ can be defined by a fuzzy aggregation operator over linguistic terms (Dubois and Prade 1985). However, this can be limitative, if complex predicates are necessary. Thus, predicates can be defined over a tree, such that the root operator must be a fuzzy aggregation operator, and the leafs can be other operators or fuzzy states described by a linguistic term. However, every leaf in this tree must be a fuzzy state.

When all rules of behavior b_i have an activation level tending to zero: $\sum_{j=1}^{n} d_{i,j} \rightarrow 0$, the cost for this behavior will tend also to zero: $f(b_i, \mathbf{x}) \rightarrow 0$. This means that there is no control of the default cost of a behavior when it is inactive. Thus, a default rule needs to be added for each behavior. This rule can be described as

$$r_{i,n+1} = k_{i,n+1} d_{i,n+1}(\mathbf{x}) \tag{4}$$

where $k_{i,n+1}$ is the value that the cost function must tend to when activation levels of other rules tend to zero. The value $d_{i,n+1}$ can be computed as

$$d_{i,n+1} = 1 - \sum_{j=1}^{n} d_{i,j}. \tag{5}$$

2.2 Goals

Goals are membership functions, g, such that $g(f(b_i, \mathbf{x}))$ is the preference level for behavior b_i

in state $\mathbf{x}$. The domain of goals must be the range of cost functions f. In this way, it is possible to have goals defined over different domains as long as there is a cost function with the same range for each goal.

2.3 Constraints

Constraints can be added to the decision algorithm. Constraints are functions, $c : b_i \to [0, 1]$, that establish a maximum preference level for each behavior. One can specify at run time a set of constraints, which represent a specific model of the world in where the robot evolves. Constraints are pieces of information extracted from the environment to contextualize the behavioral selection process.

2.4 Fuzzy Decision Making

The optimal fuzzy decision is the one that better satisfies the goals and the constraints. Thus, if g are the goals and c are the constraints, the optimal fuzzy decision is given by

$$b_{opt} = \arg\max_{b} \left[\min(g_i(f(b, \mathbf{x})), c_j(b)) \right], \qquad (6)$$

with $i = 1, \ldots, n$ and $j = 1, \ldots, m$. It is necessary to establish priorities between behaviors, because if there are two or more that have the same satisfaction level, the decision can be made by a default rule instead of being random.

2.5 Hierarchical Fuzzy Decision Making

The abstraction of relevant features is very important in complex systems. Systems with behaviors that can be decomposed in sub-behaviors reduce the complexity involved in decision making. For example, in robotic soccer each robot can play various roles, like attacker, mid-fielder, defender or goalkeeper. With exception of goalkeeper, other robots can choose roles dynamically. Each role represents a high level behavior that is composed by low level sub-behaviors. A discussion on hierarchical operators in fuzzy decision making can be found in (Kaymak and Sousa 2003).

The proposed method performs "local" fuzzy inference. Only one sub-behavior will be selected, the inference continues in this sub-level restricted to behaviors associated with the behavior chosen in the higher level. This method does not allow interference between hierarchical levels.

3. CASE STUDY ON A ROBOTIC GOALKEEPER

Under the robot soccer domain, robotic Goalkeepers are robots that are typically rich in behaviors, representing an ideal case study to test fuzzy decision making.

3.1 Problem Description

In this case study, an omnidirectional robotic goalkeeper is considered. A good starting point is to identify behaviors that human goalkeepers usually display. In a soccer glossary, see (Glossary 2006), a description of human soccer behaviors is presented. We have used some of the most typical ones for goalkeepers.

3.1.1. Goalkeeper behaviors The most important behaviors are:

- Clear Ball: kick ball to free place (can be out of field)
- Outlet Pass: pass to unmarked teammate
- Intercept ball: move to intercept ball position
- CatchBall: approach ball and catch it.
- Steal Ball: when an opponent has the ball, steal it
- Cut Down The Angle: move forward to minimize the shot angles within which the ball can get into the goal
- Marking: avoid easy pass (cut down pass line)
- Home Position: back to default position

3.1.2. States of the world For the presented behaviors the following set of world states is used:

- ballNearFinalLine
- ballMiddleField
- ballNearOwnGoal
- ballOpponentField
- ballNearGoalkeeper
- ballVelocityOwnGoal : ball velocity with respect to own goal
- ballVelocityOpponentGoal : ball velocity with respect to opponent goal
- ballVelocityStopped
- ballVelocityLow
- ballVelocityMedium
- ballVelocityHigh
- ballOwnerGoalkeeper
- ballOwnerTeammate
- ballOwnerOpponent
- ballOwnerNobody
- teammateGoodPlaceForPass : teammate is in good place to receive pass
- opponentGoodPlaceForPass : opponent is in good place to receive pass

- angMinimized : angle between goalkeeper, ball and goal post

Figures 1, 2 and 3 show some fuzzy sets describing the previously mentioned fuzzy states.

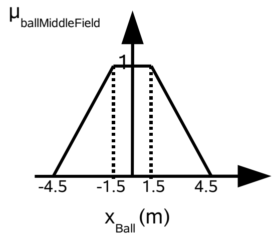

Fig. 1. Fuzzy set for linguistic term ballMiddleField (length=12, width=8 and point (0,0) is the center of field)

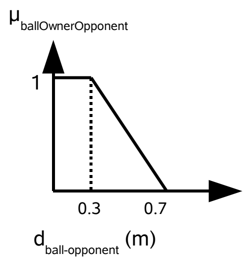

Fig. 2. Fuzzy set for linguistic term ballOwnerOpponent

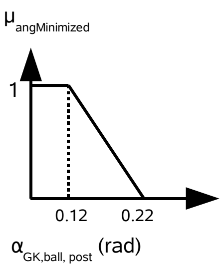

Fig. 3. Fuzzy set for linguistic term angMinimized

3.1.3. Main goal of goalkeeper The main goal of goalkeeper is to take as few goals as possible. So, this linguistic term can be defined as

$$\mu_{TakeFewGoals}(y) = \begin{cases} 1, & \text{if } 0 <= y < 2 \\ -0.5y + 2, & \text{if } 2 <= y < 4 \\ 0, & \text{if } y >= 4 \end{cases}$$

where y is the number of goals taken. The linguistic term is plotted in Fig. 4.

In this example, y is the cost. This cost is given by the cost function in (**??**) in the next section, which is applied to a specific behavior. The goal will be applied for all behaviors through the cost function.

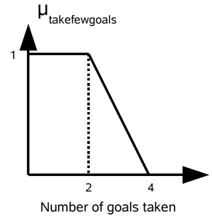

Fig. 4. Main goal of goalkeeper

3.1.4. Cost function of goalkeeper For the sake of simplicity, in this example only three behaviors will be described (ClearBall, OutletPass and CutDownTheAngle). Note however that eight behaviors were implemented. The domain of the goal is the number of taken goals per game, thus the domain of the cost function must be also the number of taken goals per game. Figures 5, 6 and 7 show block diagrams representing the behaviors, their rules, cost and priority, as defined in Section 2.1. Each behavior is represented by one or more rules. The highest priority is given to the behavior that has the minimum value. Each rule has a cost that contributes to the behavior total cost. The rule cost is determined by the product between its static cost (e.g. clearBall rule 1 has static cost 2.0) and rule activation level, that are given by the associated predicate. The main predicate has as child other predicate or a fuzzy value assigned by some fuzzy state. Each behavior has a default rule with cost 10.0, such that its activation level is $\max(0, 1 - \sum_{j=1}^{n} d_{i,j})$, where $d_{i,j}$ represents rule j activation level for behavior b_i.

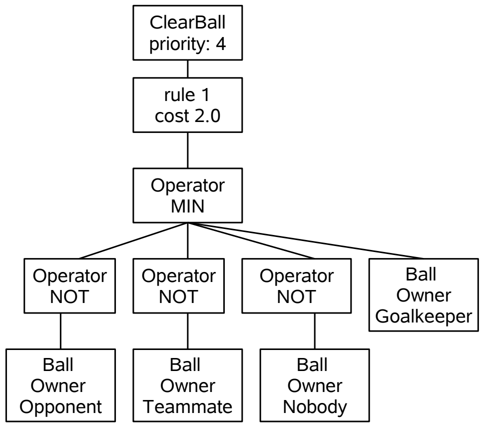

Fig. 5. Cost function structure of clear ball.

The ruler's static cost for behavior b can be defined by expertise knowledge. This cost must reflect the answer to the question: what average goals the goalkeeper take per game if execute behavior b in world state $\mathbf{x}$? For example, what must be the average number of taken goals when goalkeeper execute behavior CutDownTheAngle

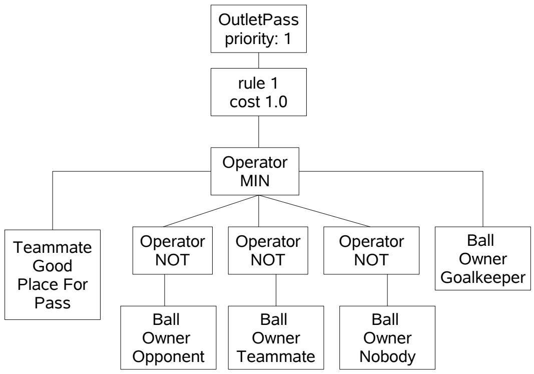

Fig. 6. Cost function structure: outlet pass

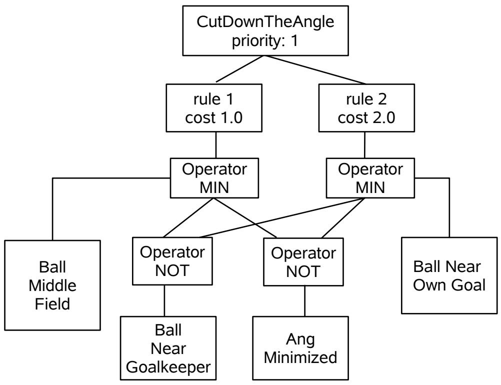

Fig. 7. Cost function structure: cut down the angle

in world state **x**. This state is composed by ball at mid-field and goalkeeper at goal. The other states are irrelevant. The expertise knowledge states that this cost must be low (one goal per game for example).

3.1.5. Constraints of goalkeeper A maximum satisfaction level is assigned to each behavior. This value is reflected in the constraints. This can be very useful for representing information concerning general information on the game state and opponents characteristics. Examples of constraints are as follows:

- $\mu_{loosing}$
- $\mu_{opponents\_have\_strong\_kick}$
- $\mu_{opponents\_kick\_up}$
- $\mu_{opponents\_dont\_pass}$
- $\mu_{opponents\_are\_fast}$

For example, when the opponents cannot pass the ball, the marking behavior should not be active. It can be simply turned off by putting all other behaviors with fuzzy value one and marking with fuzzy value zero. When the goalkeeper team is loosing, it can decide to have a more defensive behavior and constrain the level of behaviors with more risk, such as outlet pass and steal ball, to a lower activation level.

3.1.6. Fuzzy decision making applied to goalkeeper
The decision of what is the best behavior to execute in the actual world state is made by evaluating all behaviors, using the goalkeeper cost function and the "take few goals" goal. The fuzzy operator min is used to constrain the satisfaction level of goal with the maximum level allowed by constraints. The behavior that has maximum satisfaction level for the goal and for all constraints is selected. After selecting a behavior, the Behavior Coordinator orders the Behavior Executor to execute the selected behavior.

3.2 Behavior Executor

This entity does not interfere with the tasks of the Behavior Coordinator. When the coordinator decides what must be done, the executor simply does it. Next subsection describes how three of the behaviors are executed.

3.2.1. ClearBall algorithm This behavior must kick the ball to a position where there are no obstacles (opponents or teammates). Obviously, it must not kick towards its own goal. Figure 8 shows how the best choice, VC, is calculated. All vectors are sorted by angle in the field referential (from the lowest $v1$ to the highest $v7$), and the best direction to kick the ball is given by the average of two consecutive vectors that have the largest angle between them.

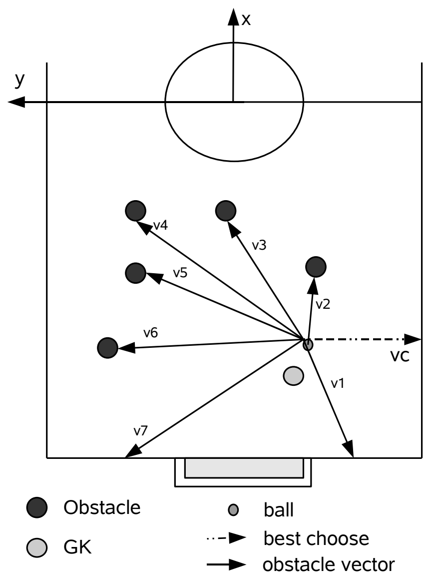

Fig. 8. Calculating the best direction to kick ball

The execution of behavior clearBall consists of performing some primitive actions, like aproachBall, rotateTheta and kickBall. With the hierarchical fuzzy decision making algorithm presented in Section 2.5, it is possible to separate the decisions. The higher level corresponds to taking a decision of what behavior must be executed, and the lower level corresponds to taking a decision of what primitive actions must be executed for each moment.

3.2.2. OutletPass algorithm With this behavior it is possible to pass to teammates that are unmarked in a good place, so that the goalkeeper also participates in attacker strategies. Figure 9

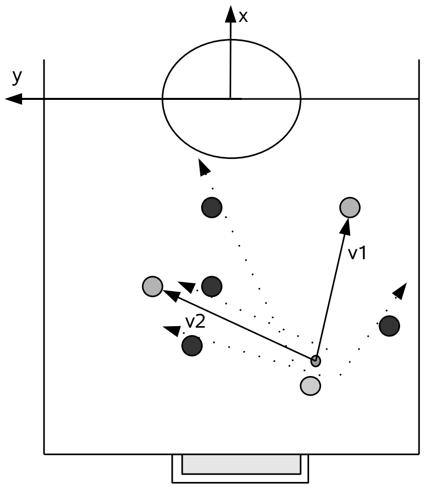

Fig. 9. Calculating the best direction to pass ball. Note that dashed vectors are not centered in ball.

shows how direction of pass is calculated. For each possibility, $v1$ and $v2$, the lowest angle between pass vector and obstacles (opponents) vectors is calculated. Direction of pass is the vector that has the largest angle. Like clearBall, outletPass is composed by aproachBall, rotateTheta and kickBall primitive actions.

3.2.3. CutDownTheAngle algorithm This behavior minimizes the angle between each post, ball and goalkeeper. By knowing the position of the ball and the diameter of the robot, it is possible to calculate the position on the field that minimizes the angles. In Fig. 10 it can be seen how to compute the position that minimizes the angles. First, one fixes the angles $t1$ and $t2$ with a value that does not represent danger. As d is known, some geometrical calculations can be applied to calculate the desired distance and the angle between robot and ball.

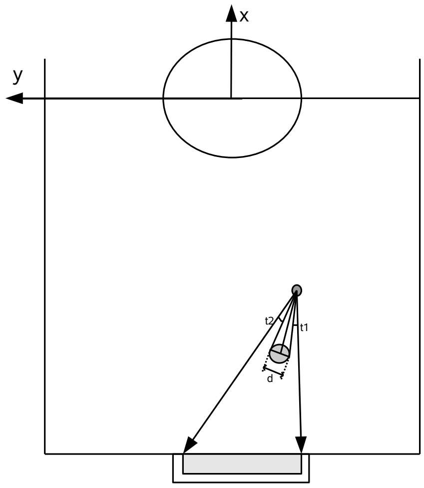

Fig. 10. Calculating the best position to cut down the angle.

4. SIMULATION RESULTS

The case study presented before was implemented in C++ into Webots simulator. Several tests were done. The simulated environment of the tests is a field with 12 m long and 8 m width, two opponent omnidirectional robots and one omnidirectional teammate. First, the goalkeeper was tested with static opponents and a teammate. All situations of the game were tested with successful and fast decision. For example, in the presence of an opponent with the ball near the goalkeeper's goal and a teammate marked by the second opponent, the goalkeeper decided to InterceptBall. When the goalkeeper is near the opponent with the ball, it decided to change for behavior StealBall. After stealing the ball from the opponent, it decided to do a ClearBall because the teammate was near to the second opponent, see Fig. 11. When the teammate moved to a good position for passing, the goalkeeper decided to make a OutletPass, but if an opponent was put between them, such that it would cut line of pass, the goalkeeper decided to make a ClearBall again.

If an opponent had the ball near the goal area and the second opponent is unmarked in danger zone with line of pass, then goalkeeper decided to CutDownTheAngle (if not minimized yet), after which it decided to execute InterceptBall. During InterceptBall, Marking is selected to cut the line of pass. After cutting the line of pass, the goalkeeper decides to intercept the ball again.

When the ball was moved to far from goal, goalkeeper decided to execute CutDownTheAngle in order to have always the angles minimized.

Our fuzzy decision making algorithm runs in 180 μs (real time) over a P4 2.6 GHz with seven behaviors. The code does not have any optimization

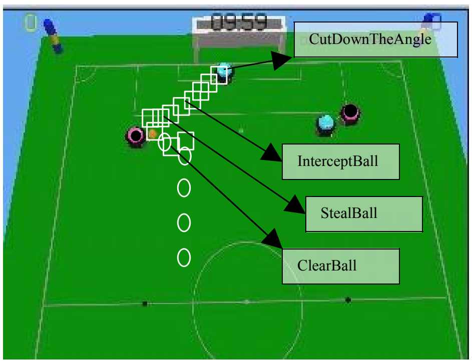

Fig. 11. Selected sequence of behaviors.

and uses generic code of STL (Standard Template Library in C++) that is usually slower than specific code. To view goalkeeper in action, please visit the site
https://omni.isr.ist.utl.pt/ nramos/fuzzyGoalkeeper/.

5. CONCLUSIONS

This paper introduced a hierarchical fuzzy decision making algorithm for behavior decision, without interference between hierarchical levels for competitive behaviors. The proposed fuzzy decision making algorithm runs fast enough for application in real robots. Designing rules for fuzzy decision is very intuitive, as shown in an example concerning a subset of the goalkeeper behaviors. This algorithm includes a novel way of representing a priori information about the environment, such as game state and opponent characteristics (e.g., slow, fast). The introduction of new behaviors does not affect the rules of older behaviors.

Future work on this framework of fuzzy decision-making will focus on adjusting the costs of rules using learning algorithms. In a first step, expert knowledge or supervised learning for quick learning can be used. To optimize the decisions without expert intervention, the next steps will use unsupervised learning with simulation. Fuzzy sets concerning fuzzy world states can be learned as well.

ACKNOWLEDGMENTS

This work was partially supported by the project POCI/EME/59191/2004, co-sponsored by FEDER, Programa Operacional Ciência e Inovação 2010, FCT, Portugal.

REFERENCES

Bellman, R.E. and L.A. Zadeh (1970). Decision-making in a fuzzy environment.. *Management Science Series* **14**(4), 141–164.

Dubois, D. and H. Prade (1985). A review of fuzzy set aggregation connectives. *Information Sciences* **36**, 85–121.

Glossary, Soccer (2006). www.firstbasesports.com/soccer_glossary.html.

Kaymak, U. and J.M.C. Sousa (2003). Weighted constraint aggregation in fuzzy optimization. *Constraints* **8**(1), 61–78.

Pirjanian, P. (1999). Behavior coordination mechanisms - state-of-the-art. Technical Report IRIS-99-375. Institute for Robotics and Intelligent Systems, University of Southern California. Los Angeles, California.

Pirjanian, P. and M. Mataric (1999). Multiple objective vs. fuzzy behavior coordination.

Sousa, J. M. C. and U. Kaymak (2002). *Fuzzy Decision Making in Modeling and Control*. World Scientific Pub. Co. Singapore.

Vadakkepat, P., O.C. Miin, X. Peng and T.H. Lee (2004). Fuzzy behavior-based control of mobile robots. *IEEE Transactions on Fuzzy Systems* **12**(4), 559–565.