J. Basic. Appl. Sci. Res., 3(11)439-446, 2013
© 2013, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Player Level Decision Making of Humanoid Soccer Robots Using Fuzzy Rule-Based Classification Systems

Pouyan Nasrollahi[1], Shahram Jafari[2], Mohammad Bazmara[3*], Alireza Nikooee[4]

School of Electrical and Computer Engineering, Shiraz University

Received: September 3 2013
Accepted: September 22 2013

ABSTRACT

After the progress in Basic Skills such as walking, turning and stability control in humanoid robots, one of the most important issues in humanoid soccer robots is the ability to decide as a player and choosing the best action to reach the destination and using the best sequence of actions in different cases. This article presents a new method for player level decision making and choosing the best action and fast ball possession via FRBCS (Fuzzy Rule-Based Classification Systems) and Rule Based expert systems. Choosing the best action based on playground information in each cycle of the game, would result in choosing the best combination of actions to reach the destination in the least possible time.

In this article, we save the game information and the best combinations of actions via putting the pall in different situations and then controlling and choosing the best robot action in each moment. Afterwards, we convert this information into rules using Fuzzy Rule-Based Classification Systems. The robot's decision will be done via importing the obtained "Rules" to the expert system.

KEYWORDS: Humanoid Soccer Robots, Player level Decision Making, Expert Systems, Fuzzy Rule-Based Classification Systems

I. INTRODUCTION

As far as football is a dynamic game, it has been the robotics' main concentration. Recently, after the humanoid robots' creation, humanoid soccer robots come to exist for robotics science and generally the Artificial intelligence science development [1, 2]. At the beginning of humanoid soccer robots' existence, the main challenges were the issues such as waking, turning and stability control while walking and turning. Generally at the end of 1960, ZMP presented a dynamic stability as a point on the ground, in which the total torque created by connections to the ground is equal to zero [3]. Afterwards, some "Stability Concepts" were presented such as FRI(Foot rotation indicator) [4] and VSR(*Valid Stable Region)*[5]. After the successful presentation of the basic skills and stability methods in humanoid robots field and the lower layer progress, the robot's ability to use basic actions in making "Player level Decision" and choosing the appropriate action in different situations in the game is the main challenge at the next step (the intermediate layer).

By stating "Player level Decision" we mean the choosing of the best actions among possible actions in every situation and combining those actions to reach a specific point on the ground or behind the ball, That the artificial intelligence offers different methods which are implemented in lots of fields such as : soccer talent exploration, and it's goalkeepers truly [6,7], or the systems of choosing outstanding features [8] and also the Geographical Information Systems that benefit from the Fuzzy uncertainty[9] and moreover using the Genetics Algorithm in Path Planning for planning the robot's motion without colliding into the objects around it and finding the best path among the possible paths in a Planner Robot or Designing an effective joint controller for periodic motion, and predefined trajectory tracking to achieve accurate tracking of desired motion in tow link robotic manipulator using MLP neural network[10,11]. Also some decision making approach in robotic field was proposed such as new walking model was named omni directional walking and could perform turn action during walking and enabled the robot fasting arrive to ball and better performance and decision making[18] and hierarchical reactive control method for humanoid soccer robots[19] and new approach for learning footstep prediction from motion capture[20].

There were some attempts to classify and choose the best action in Player level Decision such as action classification using the Classification Algorithms and Machine Learning [12], Actions Classification in the mobile robots using ANNs [13], Action Classification using tracking the human's body parts via a hierarchical Algorithm (Hierarchical Annealing Particle Filter-H-APF) [14], and Actions Classification via Evidence Feed Forward Hidden Markov Model (EFF-HMM) [15]. The issue of and Actions Classification in both humanoid robots and Human beings is almost the same but it has also some differences due to the robot's view limitations, robot's lower joints flexibility, which result in the robots movement limitations compared to a human being. Moreover, the robot's different states would be defined via a limited sequence of parameters. As it is stated in

[*]**Corresponding Author:** Mohammad Bazmara , School of Electrical and Computer EngineeringShiraz University.

the previous researches, the main concentration was more the Human Actions Classification instead of the humanoid robots.

In this article, we present a Decision Engine at the player level and the appropriate combination of actions via creating "Fuzzy Rules" and the use of these rules in the Expert System for humanoid robots in SimSpark environment which is an open text simulator. In part 2 we will explain about the "Data Collection" method. Then in the next part the Decision Model will be defined. In Part 4 the decision making model will be evaluated and finally the last part is all about the obtained results and the conclusions, based on the model.

II. METHODOLOGY

- **Simulator and Environment**

The simulation Environment was originally created to foster research on soccer game playing at a higher level than was previously possible in the real world because of the contemporary restraints of robotics technology.

While it started out as a two dimensional simulation, later on the 3D simulation competition (2004) improved on this by adding an extra dimension. Thereby increasing the realism of the simulated environment and making it more comparable to other (real world) Robocop league environments; but still being able to simulate more players on the field than would be practical in, say, the middle size league or even the small size league. Thus giving rise to higher level research on larger groups of slightly more realistic soccer playing robots. But it were still only sphere shaped agents running around the field.

Finally as of 2006, SimSpark, a generic physical multi-agent simulator system for agents in three-dimensional environments simulates 3D models of humanoids, based on robots used in the non-simulated Robocop humanoid league. This opens up opportunities for research on higher level control of humanoid robots as well as higher level behaviors in humanoid soccer and getting still closer to how humans play the game. Now SimSpark is used as the official Robocop 3D simulation server [16] and simulates virtual NAO robot with 22 degrees of freedom, (shown in Fig. 1).

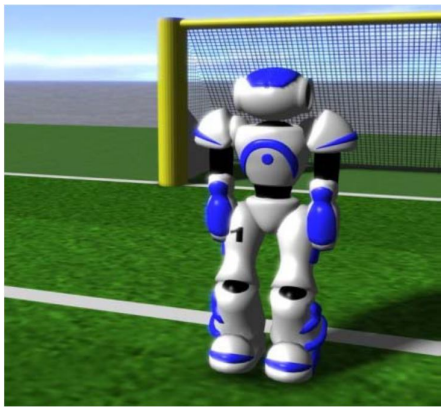

Figure 1. Image of virtual Nao robot

- **Data Collection**

After putting the robot on the ground, an operator starts Data Collection and making a dataset using a Toolkit Trainer, which is a tool for commanding the robot and is shown at the figure 2.

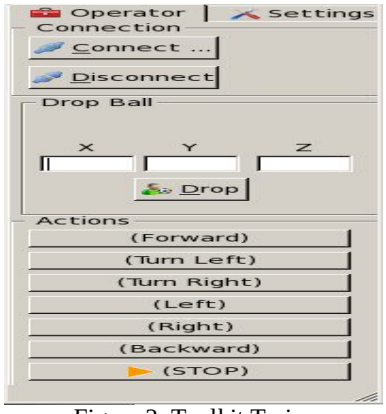

Figure 2. Toolkit Trainer

As seen, the made Toolkit consists of 3 independent parts. The **Connection** part connects and disconnects the connections to the simulation server. The **Drop Ball** part is responsible for dropping or throwing the ball to different points on the playground randomly. The **Action** part helps the operator choose the appropriate action and command the robot. First the operator connects to the server, using Connection part and then Beams the ball to a random point on the playground using Drop Ball section. After the ball is placed on a random point, the operator should combine some actions to reach the ball, just like a computer game controller. As an action is chosen, the robot should perform that action and the robot's properties, ball, environment and also the current action's number will be logged into the dataset. While switching between the actions, the operator uses STOP button, which is used for stopping the logging operation and thinking about the next action.

After dropping the ball to the different points randomly for several times and moving the robot behind the ball, a huge dataset of Decisions (from Human's point of view) based on the mentioned method will be created (figure 3).

1	Distance to ball	Theta to ball	Ball pos x	Ball pos y	My pos x	My pos y	Theta to opp goal	Distance to opp goal	CurrentSkill
2	0.685493	74.3089	10.7121	-9.60228	10.5	-10.2541	74.3089	0.685493	0
3	0.673498	73.6686	10.6948	-9.59942	10.5	-10.2441	73.6686	0.673498	0
4	0.661438	74.6278	10.6791	-9.5974	10.5	-10.2341	63.7118	9.24901	0
5	0.637103	77.0881	10.6534	-9.61576	10.5	-10.2341	63.9594	9.23399	0
6	0.62482	79.3978	10.6387	-9.59488	10.5	-10.2041	79.3978	0.62482	0
7	0.649307	82.0262	10.625	-9.54695	10.5	-10.1841	66.8422	9.23399	0
8	0.661438	85.1881	10.5814	-9.58772	10.5	-10.2441	67.3731	9.19394	1
9	0.649307	87.6021	10.571	-9.58872	10.5	-10.2341	68.89	9.17892	1
10	0.649307	84.4162	10.5497	-9.60673	10.5	-10.2541	84.4162	0.649307	1
11	0.637103	84.1305	10.5482	-9.54883	10.5	-10.1841	62.1773	9.18393	1
12	0.637103	84.1305	10.5482	-9.54883	10.5	-10.1841	84.1305	0.637103	1
13	0.637103	84.1305	10.5482	-9.54883	10.5	-10.1841	84.1305	0.637103	1
14	0.637103	83.5362	10.5401	-9.58828	10.5	-10.2241	60.5009	9.17892	1
15	0.637103	83.5362	10.5401	-9.58828	10.5	-10.2241	83.5362	0.637103	1
16	0.637103	83.5362	10.5401	-9.58828	10.5	-10.2241	83.5362	0.637103	1
17	0.649307	84.402	10.5465	-9.57648	10.5	-10.2241	61.9321	9.20395	1
18	0.649307	84.402	10.5465	-9.57648	10.5	-10.2241	84.402	0.649307	1
19	0.649307	84.402	10.5465	-9.57648	10.5	-10.2241	84.402	0.649307	1
20	0.661438	82.5328	10.5417	-9.53399	10.5	-10.1941	59.7629	9.21396	1

Figure 3.dataset of Decisions from Human's point of view

As shown in figure 2, the table has 9 columns. The 1st column is distance between the ball and the robot. The 2nd column shows the robot's body angle to the ball. The 3rd and the 4th column show the current X and Y of the ball on the ground. The 5th and the 6th column indicate the X and Y of the robot on the ground. The 7th column shows the robot's body angle to the opponent's goal. The 8th column indicate the robot's distance to the opponent's goal and finally the 9th column shows the type of appropriate skill for the robot, which is explained in detail in Table 1.

Table 1. Description for the Considered Skills for the robot

#of action	Action's Name	Actions Description
0	FORWARD_WALK	The proceeding of robot forward
1	TURN_LEFT	The turning of robot to the left side
2	TURN_RIGHT	The turning of robot to the right side
3	LEFT_WALK	The movement of robot to the left side
4	RIGHT_WALK	The movement of robot to the right side
5	BACKWARD_WALK	The movement of robot Backward

- **Robot Decision Making**

As stated before, the data are gained serially and because the robot is moving forward most of the times, the data in many cycles will have the value of the class "FORWARD_WALK ", which makes the data unbalanced. To solve this problem, a method is presented in which we use the Critical and Effective data. The Critical Data is the one which is continuous around the skill changes. In this attempt, we choose a number of data (α) before and after the skill change as a critical data.

After the first level, FRBCS that is a popular classifier system and representing the classifier with a set of fuzzy if-then rules is used to construct decision making rule base from obtained dataset. To achieve this, we convert these data progression into fuzzy data then we generate some rules and assign them a weight using those data. Afterwards we use the created rules to classify the data. The process of making a Decision Engine has 4 parts: fuzzification – rule generation – weighting – Data Classification.

- **Fuzzification**

For this step we should normalize the data first using the Formula 1, so that the data's value would be varied between 0 and 1.

$$Data = \frac{(Data - minData)}{(maxData - minData)} \quad (1)$$

Where "minData" is the lowest value in each feature of the Training Data and "maxData" is the highest value. Then we define 14 Membership Function on each feature and one "Don't care" membership Function which has the amount "1" for all of the values (Figure 4).

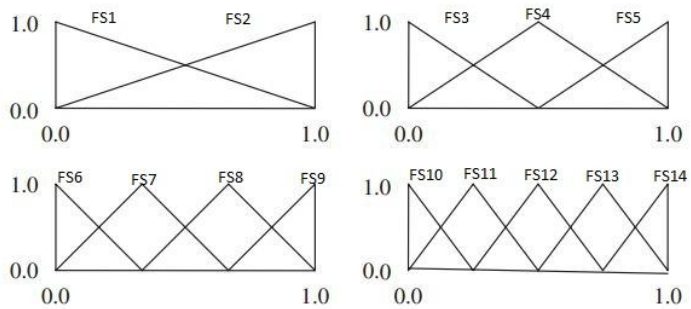

Figure 4. Membership Function on the fuzzy set defined on the data[17]

- **Rule Generation**

If we want to make all of the "antecedent- combination", we need $14^{|Feature|}$ rules and when the number of features gets higher, Then the number of rules would increase Logarithmically. Therefore we only create the rules with the length of 1 or 2, so the possible situations can be calculated via Formula 2.

$$\binom{n}{2} * 14^2 + n * 14 \quad (2)$$

Where "n" is the number of features and we use "Confidence" which is related to the supported bound of each rule to create "Consequent" for each rule as the Formula 3.

$$Confidence\left(A_j \Rightarrow ClassC_j\right) = \frac{\sum_{X_p \in ClassC_j} \mu_j(X_p)}{\sum_{p=1}^{m} \mu_j(X_p)} \quad (3)$$

$$q = \underset{1 \le h \le M}{\arg max} \left\{ Confidence\left(A_j \Rightarrow ClassC_h\right) \right\} (4)$$

Where μ_j is the Compatibility Grade for type X with antecedent rule "j" and "q" chosen class for the rule j. yet, the number of rules is huge therefore some of the "better" rules of each class will be kept and we create "Rule Base". The evaluation measure for the rules would be graded based on Formula 4:

$$e(R_j) = \sum_{X_p \in ClassC_j} \mu_j(X_p) - \sum_{X_p \notin ClassC_j} \mu_j(X_p) \qquad (5)$$

- **Data Classification**

After the Rule Bass is created, we should classify the new data, using Rule Base. For this classification, there are 2 ways.

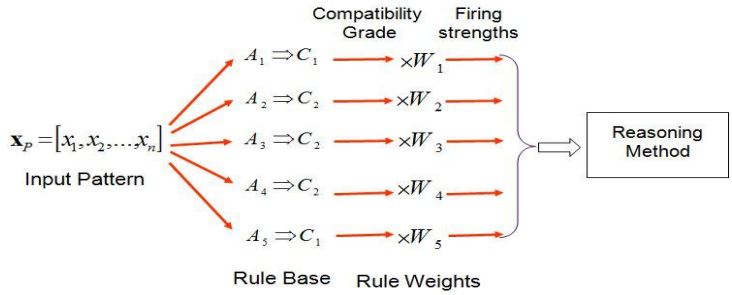

Figure 5. The process of Classifying a pattern

Single Winner: the class assigned to "Pattern X" is equal to Consequent of the rule which hast the higher "Firing Strength" for this pattern (figure 5).

Weighting Vote: in this method the total amount of the "Firing strength" for each class is calculated and the class which its Firing strength is higher, will be assigned to "Pattern". Finally In our example we used the single winner approach.

- **Rule Weighting**

For system optimization, we may assign some appropriate weights to the obtained rules from previous steps. We do this for all of the rules and the results may be more accurate by repeating this assignment. The process of choosing the best weight for a rule has 3 steps:

1- First, we find the critical dataset. For this, we set the value of the weight of the desired rule to 0 and classify all of the "Training Data". Then we consider a very huge amount (a very large number) for the value of that rule and classify all of the "Training Data" again. The critical data is correctly classified only in either of the mentioned states.

2- Then the critical data's "score" will be calculated and if the rules' weight is equal to the amount of "score", this data is located on the borderline of the classification for the current class and all of other classes. Formula 5 shows the calculation for 2 states (Single Winner and Weighting Vote):

Single winner:

$$S(X_t) = \frac{\varphi(X_t)}{\mu_k(X_t)} \quad (6)$$

$$\varphi(X_t) = \max_{1 \le j \le N} \{CF_j . \mu_j(X_t) | R_j \epsilon S,\ Consequent(R_j) \neq ClassT\} \quad (7)$$

Weighted vote:

$$S(X_t) = \frac{\Psi(X_t) - \sigma_{ClassT}(X_t)}{\mu_k(X_t)} \quad (8)$$

$$\Psi(X_t) = \max_{1 \le Q \le M} \{\sigma_{ClassT}(X_t) | Q \neq T\} \quad (9)$$

$$\sigma_{ClassT}(X_t) = \left\{ \sum_{\substack{R_j \in S, \\ C(R_j) = ClassT}} \mu_j(X_T) CF_j \right\} \quad (10)$$

At the 3rd step, we calculate the average score of 2 sequential "critical data" and "classifier accuracy" each time for the training data. Then we specify the best accuracy and set the rule's weight equal to the average conforming accuracy for all of training data.

The 3rd step should be followed for 3 times iteratively:

The important issue is that the weight of rule with $max(scores) + \varepsilon$should be checked in addition to the average score of all critical pairs of data. Moreover, if a rule has no critical data, the weight for that rule is equal to ZERO.

III. Evaluation & Result

The In the basic sample, almost 256000 data for training were extracted from the software. First, we assign the data to the desired classes, using the proposed method. The following table compares the proposed method with the other classifiers for this problem:

Table 2. Description for the Considered Skills for the robot

Method	Train Accuracy	Test accuracy (10-fold)
Proposed method	73.8	72.3 (± 1.3)
Neural Network (2layer MLP)	72.62	70.96 (± 2.2)
C4.5	68.41	65.96 (± 3.2)

To evaluate the proposed method, the sequence of suggestions that the expert system suggests to the robot, should be compared with those suggested by an expert person.

The considered experts (people) are: 2 experts in simulator competitions (Members of Delta3D team), one professional league footballer, 2 semipro footballers and one regular person. For this, the "footballer" and "ball" agents each one is simulated in a specific point on the playground individually and randomly. Then the stated expert system and the expert people should suggest a combination of appropriate actions to reach the ball, considering the agent's angle to the opponent's goal. After each action is suggested, the agent will perform the

specified action as chosen until it reaches the point behind the ball and the sequence of the actions will be terminated.

The presented experts send a sequence of actions by which they can guide the soccer agent to the desired points on the playground, using “Toolkit Trainer”. In fact, the experts can command the robot using 6 different actions as desired to move the robot to the destination. we do this for 100 times and in each time the random agent’s coordinate and the ball’s coordinate will be announced to the experts for giving the values to initial ball and agent’s coordinate and after choosing the first action by an expert, the timer will start automatically and when we move the robot to the ball, considering the appropriate angle to the opponent’s goal, the expert will stop the timer and the sequence of the actions by pressing “STOP” which is located in the Toolkit Trainer. The main thing is that the duration between the time when the agent reaches the point behind the ball and pressing the button “STOP” is almost nothing and has no effect on the suggested sequence that the expert suggested. For example in the figure 6 the cases when the expert guides the robot to back is illustrated.

Figure 6. Expert’s recognition of backward Movement

The considered evaluation measures include the average ball reaching time and the final robot’s average Straight to the opponent’s goal in 100 iterations. Straight is the angle difference between the agent to the ball and the agent to the opponent’s goal.

The next evaluation measure consists of the time to reach the ball and the required time to reach the “ZERO” Straight.

Table 4. The Obtained Results comparing Experts, The Previous system and the Expert system

Examiner	Time total	Angle (deg)	Time dis (sec)
Expert Sys.	40.1	31(±12)	38.4(±15.8)
Expert 1	39.2	32(±13)	36.8(±22.3)
Expert 2	41.6	36(±10)	38.2(±23.2)
Footballer 1	48.3	34(±14)	45.2(±16.7)
Footballer 2	50.3	37(±13)	46.9(±20.5)
Footballer 3	52.5	40(±12)	48.5(±22)
Regular Person	61.95	30(±18)	59.4(±16.8)
Delta3d 2011	81.2	7(±3)	80.4(±28.8)

By reading the table above, it can be concluded that although one expert did better than the suggested system, an acceptable progress in the ”expert system” can be observed comparing to the other systems such as “Delta3D” implemented in 2011.

The final evaluation illustrates that our proposed method reduced the number of falls and action switches in 100 iterations.

Table 5. Impact of desiction proposed method in robot’s performance

Examiner	Average Number of action changes	Average Number of robot falls
Expert Sys.	23	2
Expert 1	29	6
Expert 2	33	8

IV. Conclusion

As far as fast reaching behind the ball is considered as the team's advantage in the humanoid soccer robots Competitions, We could improve robot's performance by adding mid-level decision making and choosing the best skill in each moment while the robot attempts to reach the point behind the ball on the playground.

As we have shown, we could generate the required data and information in a new and trustful way and could obtain their valuable properties and create dataset. Then, after creating the effective dataset, we could extract the best and the most optimized rules from those data, by which we could observe that the rules are very accurate in the "Evaluation" section of the article.

Finally by importing the stated datasets into the Rule BasedExpert Systems, we created a strong Inference and Decision Engine with a very new and different method compared to Decision and Conditional systems based on "if–else " for the Player Level Decision in the humanoid soccer robots with a high rate of response by decreasing number of turn actions and optimal combination of low level actions.

The experimental work of this research demonstrated the fact that if data mining and machine learning techniques will be used with an appropriate dataset they will be effective enough to be considered alongside other usual approaches in robotics. Our next analysis will be focused on considering conditional parameters provided by the simulation server and enlisting the opponents by observing their spatio-temporal information and deal with them as high-level decision making.

Acknowledgment

The authors declare that they have no conflicts of interest in this research.

REFERENCES

1- H. Kitano, & M. Asada, 2000. The RoboCup humanoid challenge as the millennium challenge for advanced robotics, *Advanced Robotics, 13*(8): 723-736.

2- C. Zhou, 2004. Rules for the RoboCup humanoid league, *Advanced Robotics, 18*(7): 721-724.

3- M. Vukobratovic, D. Juricic, 1969. Contribution to the synthesis of biped gait, IEEE Transactions on Bio-Medical Engineering, 16(1): 1-6.

4- Goswami, Foot rotation indicator (FRI) point: A new gait planning tool to evaluate postural stability of biped robots, Proc. IEEE International Conference on Robotics and Automation, Detroit, MI, 1999, 47-52.

5- Q. huang, K. Tanie and S. Sugano,2000. Coordinated motion planning for a mobile manipulator considering stability and manipulation, IEEE International Journal of Robotics Research, 19(8): 732-742.

6- M. Bazmara, S. Jafari and F. Pasand, 2012. A Fuzzy expert system for goalkeeper quality recognition, International Journal of Computer Science Issues,9(5): 36-42.

7- M. Bazmara, S. Jafari, 2013.K Nearest Neighbor Algorithm for Finding Soccer Talent, Journal of Basic and Applied Scientific Research, 3(4): 981-986.

8- Jamali, M. Bazmara, and S. Jafari, 2012. Feature selection in imbalanced data sets, International Journal of Computer Science Issues, 9(3): 42-45.

9- F. Mohammadi, M. Bazmara, 2013. A New Approach of Fuzzy Theory with Uncertainties in Geographic Information Systems, International Journal of Mechatronics, Electrical and Computer Technology, 3(6): 1001-1014.

10- S. H. Hashemipour, H. Karimi, A. Adeli, 2013. Neural Network MLP with Sliding Mode Controller for Robotic Manipulator, Journal of Basic and Applied Scientific Research, 3(1): 512-520.

11- H. Yazdani, A. Fallah and S. MahmodHoseini, 2012. A new approach for robot Path Planning with Genetic Algorithms, Journal of Basic and Applied Scientific Research, 2(4): 4122-4129.

12- P. Nasrollahi, S. Jafari, M. Ebrahimi, & M. Ebrahimi, Action Classification of Humanoid Soccer Robots Using Machine Learning, 16th CSI International Symposium onArtificial Intelligence and Signal Processing, Shiraz, Fars, 2012, 598-603.

13- T. Theodoridis, & H. Hu, Action Classification of 3D Human Models Using Dynamic ANNs for Mobile Robot Surveillance, *Proc. ROBIO'07*, 2007, 15 -18.

14- L. Raskin, M. Rudzsky, & E. Rivlin, 3D human body-part tracking and action classification using a hierarchical body model, Proc. British Machine Vision Conference, 2009, 12.1-12.11.

15- M. Del Rose, C. Wagner and P. Frederick, 2011.Evidence feed forward hidden Markov model: a new type of hidden Markov model, International Journal of Artificial Intelligence & Applications, 2(1): 1102.0899.

16- Wiki Page. (2012) The SimSpark Website. [Online]. Available: http://simspark.sourceforge.net/wiki

17- H. Ishibuchi, T. Yamamoto, 2004. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, Fuzzy Sets and Systems, 141(1): 59–88.

18- P. MacAlpine, P. Stone, 2012. Using Dynamic Rewards to Learn a Fully Holonomic Bipedal Walk, AAMAS Adaptive Learning Agents Workshop:1-8.

19- S. behnke, J. Stucker, 2008. Hierarchical Reactive Control for Humanoid Soccer Robots, International Journal of Humanoid Robotics, 5(3): 375–396.

20- A. Schmitz, M. Missura and S. Behnke, Learning footstep prediction from motion capture. In *RoboCup 2010: Robot Soccer World Cup XIV* (pp. 97-108). Springer Berlin Heidelberg.