

Hierarchical Case-Based Reasoning Behavior Control for Humanoid Robot

BASSANT MOHAMED EL-BAGOURY, ABDEL-BADEEH M. SALEM,
AND HANS-DIETER BURKHARD

ABSTRACT. Behavioral control for an autonomous humanoid robot is a very complex problem, especially in the RoboCup domain. This is due to the dynamics of the environment and the complexity of behaviors that should be executed in real-time. This paper presents a new behavioral control model that depends on case-based reasoning. It is applied for humanoid soccer robot. It consists of a hierarchy of four levels, the first level is to decide robot role as attacker, the second level is decide which skill to execute as goal-scoring, dribbling or pass, the third level is to determine the behaviors of each skill as kicking or walking and the fourth level is to adapt lower-level behaviors as distance to walk and angle of kick. The behavior control is implemented in the framework of Webots Simulation tool for (NAO) humanoid robot. We achieve an average good accuracy that reaches 86.5 % .

1. Introduction

Humanoid Robots are the most recent development in intelligent robotic control and autonomous agents [1] . There are more than major humanoid robot projects around the world as Honda [2]. Humanoid Robots requires artificial intelligence AI [4] techniques to act autonomously in dynamic and complex environments.

The RoboCup [5] competition promotes AI and robotics for technologies research. RoboCup Humanoid League is one of the most dynamically progressing leagues. Our NAO Team Humboldt [1] is developing a complete project to participate in RoboCup World Championship 2009 [2]. Our research developments include motion planning, image processing, robot control and behaviour-based control [1].

Behavior-based control [5] has become one of the most popular approaches to intelligent robotics control. The robot's actions are determined by a set of reactive behaviors, which map sensory input and state to actions. Despite of the behavior-control part, most of robotics systems use classical behavior-control architectures. These classical architectures can cover all sensory input states of complex environments and thus limits the robot ability to adapt its behaviors in unknown situations. Recently, some AI techniques such as neural networks, fuzzy logic have been applied successfully to behavior-control of mobile robots [5]. However, research on behavior control of humanoid robots using AI is still in initial stage [6].

Case-Based Reasoning (CBR) is a reasoning methodology that simulates human reasoning by using past experiences to solve new problems. New problems are solved by retrieving and adapting the solutions to similar problems [7]. The most crucial CBR tasks involve; case indexing, representation, retrieval and adaptation. In robotics, CBR has been applied recently for many robotics types and tasks. For example, Arcos et al., [8] uses CBR for autonomous mobile robot navigation. Urdiales [9] presents a new reactive layer that uses CBR for robot navigation. CBR has also been widely applied in the RoboCup domain [10].

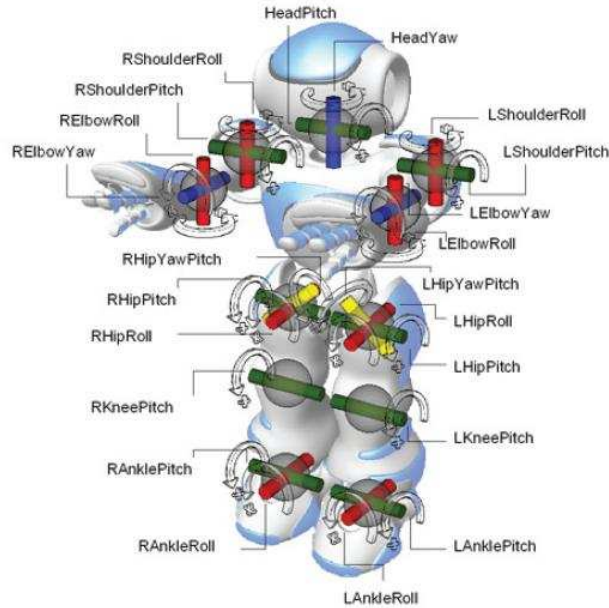


FIGURE 1. NAO Humanoid Robot

We also develop a CBR behavior control of NAO humanoid robot [11]. However, our previous CBR architecture was simple and it does not implement a complete adaptive behavior controller. In this paper, we extend our previous work [11] and we propose a new CBR behavior controller for NAO humanoid robot. Its reasoning depends on the hierarchical CBR technique, which enables our NAO robot to reason about complex behaviors by adapting multiple cases at various levels of abstraction. Our main goal is to develop a adaptive behavior controller that will be able to reason about more complex behaviors as dribble and goal-score. The coming sections describe our main architecture and results.

2. Webots for NAO Humanoid Robot

Our work is currently being implemented in the framework of Webots simulation for the NAO humanoid robot [1]. Webots is developed by Cyberbotics Ltd. It is simulation software for modelling, programming and simulating robots. It is also an open source tool and all code can be modified by visual studio using C++ programming language. NAO humanoid robot is developed by Aldebaran Robotics [12]. It has been officially selected to replace AIBO in RoboCup competition. It is an excellent robot in terms of mobility, performance and reliability. As shown in Figure 1, it has 22 degrees of freedom.

3. Hierarchical CBR Behavior Control

This section presents our proposed adaptive behavior-control for NAO Humanoid Robot. It depends on CBR technique to simulate the reasoning process of soccer players and to adapt (NAO) behaviors as goal-score and dribble in real-time RoboCup

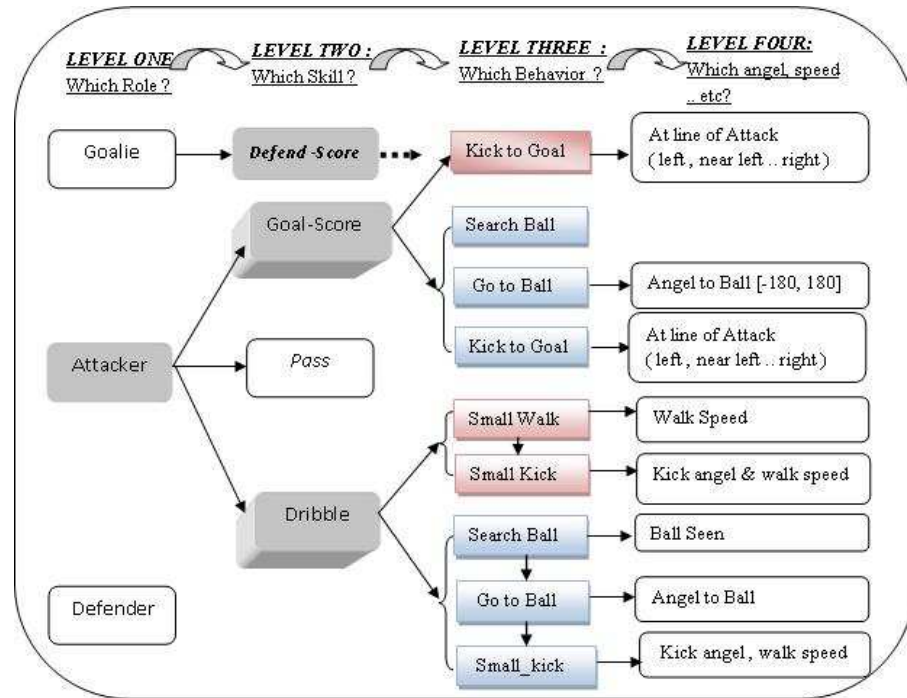


FIGURE 2. Hierarchical Behaviors Scheme of our CBR Control

dynamic environment. The coming subsections describe in details the main behaviors of RoboCup and our proposed behavior-control.

3.1. NAO Behaviors. In the RoboCup domain, soccer behaviors are complex behaviors as dribble and goal-score. However, in our project of NAO team Humboldt [1], we realize behaviors using simpler ones. For example, the goal-score behavior consists of kick-ball if ball is seen or search-for-ball, go-to-ball and kick-to-goal if ball is not seen. The dribble behavior consists of small-walk and small-kick then search-for-ball, fast-walk-to-ball, small-walk and then small-kick. In order to reduce the overall complexity of behaviors implementation, we present a new scheme of our NAO behaviors. As shown in Figure 2, it consists of hierarchy of four phases. The first phase determine robot role, the second determines robot skill, while the third phase for robot basic behaviors and the fourth phase finally for details implementation of behaviors.

3.2. HCBR Behavior Control Architecture. Our main goal in this research is to develop an adaptive behavior-control for NAO. The controller should execute complex soccer behaviors in real-time. Also, it should adapt these behaviors according to the current game situation. This section describes our new hierarchical case-based behavior controller. Its main architecture is shown in Figure 3. As shown, it simulates our soccer behavior scheme as described in the previous section (section 3.1). It consists of a hierarchy of four levels. Each level has a specific task. The first level is to decide robot role as attacker, the second level is decide which skill to execute as goal-scoring, dribbling or pass, the third level is to determine the behaviors of each skill as kicking or walking and the fourth level is to adapt lower-level behaviors as

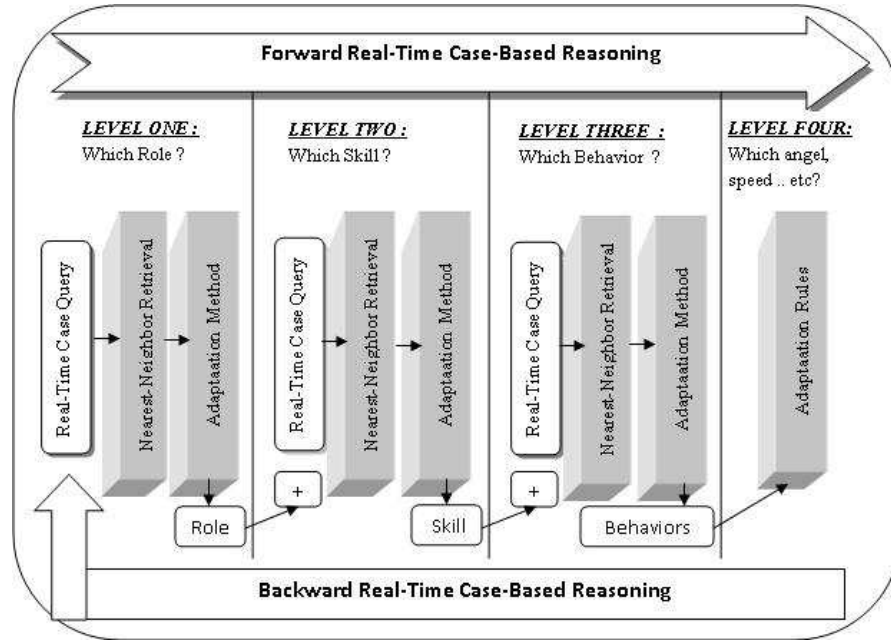


FIGURE 3. Hierarchical CBR Behavior Control Architecture

distance to walk and angel of kick. The coming subsections will describe our CBR engine in detail.

3.2.1. Case Representation. CBR engines are based on having a case-memory [7]. A case-memory acts as the main memory of experience for an agent to reason. A case-memory should be robust and covers a domain. A lot of research is done concerning case-memory organization and maintenance. Each case-memory consists of a set of cases. A case represents a single experience that an agent has experienced before. There are different case representations schemes as flat or object-oriented [7]. For our NAO robot, its case-memory represents its accumulated experience that he learns in different RoboCup game situations. A case in NAO case-memory represents a single situation that he has experienced before. For example, a sample case experience can be defined informally as:

Sample Case 1: A NAO robot acts as an attacker player and he score a goal in left direction and it was successful.

Another more complex case is informally described as:

Sample Case 2: A NAO robot acts as acts as an attacker player and he scores a goal in left direction and it was not successful because another robot who acts as goalie defend the ball.

These two situations look very similar but in fact, they are different experience and the decision that should be done by NAO CBR engine is totally different. NAO should adapt its behaviour of normal goal-scorer to execute in case if goalie is expected to defend the ball. Each case in our behavior control is represented in a flat-scheme , where each case consists of problem features and solution features. This is considered in our engine as a single concrete case. In order to support our hierarchical architecture, we further decompose each concrete case into a set of abstract cases. As shown

in Figure 4, each concrete case consists of four abstract cases. The first is called role-sub-case, where each case consists of problem features and solution features. This is considered in our engine as a single concrete case. In order to support our hierarchical architecture, we further decompose each concrete case into a set of abstract cases. As shown in Figure 4, each concrete case consists of four abstract cases. The first is called role-sub-case, it consists of features of robot role and the abstract solution of level one. The second is called the skill-sub-case, it consists of features and abstract solution of robot skill. The third is called the behavior-sub-case, it consists of features and abstract solution of robot behaviour. Finally, the last sub-case is called the adaptive-sub-case, it determines the final detailed overall case behaviour solution. Our main goal of this decomposition is to reduce the overall problem complexity by solving smaller sub-problems. Figure 4 shows a sample of our case representation.

3.2.2. Case Retrieval and Forward Reasoning. NAO behaviors should be executed in real-time soccer game. For this, we apply its CBR engine in two directions, forward and backward. This will enable NAO to continuously adapting its behaviours from case (state) to case (state). For example, after executing dribbling behaviour it can automatically backtrack to execute the goal-score behavior. A single behavior of NAO is determined using the forward reasoning path. Given a new real-time case query of NAO, it will be decomposed into three sub-cases, which are the new role-sub-case, the new skill-sub-case and the new behaviour sub-case. For each new sub-case, the Nearest-Neighbour retrieval algorithm [7, 11, 13] is used to determine its most similar sub-case from the case-memory. It computes the similarities between two sub-cases as the weighted average of their local similarities of features. The global similarity function is computed as follows:

$$Similarity(N, R) = \sum_{i=1}^n Sim(N_i, R_i) \times W_i \quad (1)$$

This function takes any value in the similarity range from 0% to 100%, where 0% is total dissimilarity and 100% is an exact match. \mathbf{N} is the new sub-case and \mathbf{R} is its corresponding retrieved sub-case. \mathbf{W}_i is the weight of feature i that reflects its importance on the diagnosis decision of the current new sub-case. Sim is the local similarity functions, which we use computed similarity between local case features. We use two types of these functions. A Boolean function to compute local similarity of Boolean features and it is defined as:

$$Sim(N_i, R_i) = 1 / \{1 + |N_i - R_i|\} \quad (2)$$

In our system, feature weights are initially assigned by making analysis to the cases in the case-memory. All weights are normalized in the range [0-1]. All the critical features of each sub-case are used as indexes to speed up the retrieval process [7].

3.2.3. Case Adaptation and Backward Reasoning. NAO behaviour solution is solved by case adaptation. Case adaptation is performed in a top-down fashion using our hierarchical case-based behaviour control. Case adaptation starts with Level-One, which it adapts Abstract-Solution1 to decide the NAO role as attacker or goalie, while Level-Two adapts the Abstract-Solution2 to decide NAO skill as goal-score or dribble. The Level-Three adapts the Abstract-Solution3 to decide NAO behaviours as kick to goal. Finally, NAO executes its real-time adapted low level behaviours in Level-Four, where it adapts low-level values such as angel-to-kick and speed-of-walk.

	Feature No.	Case Feature	Value
Level one	1.	Case_Index	1
	2.	Robot_Role	1 (attacker)
	3.	Robot_Color	2 (red)
	4.	Robot_angel	0.0000189
	Abstract Solution_1 Level One	Abstract_solution1	Attacker
Level two	5.	Case_Index	1
	6.	Abstract_solution1	Attacker
	7.	Robot_x	1432
	8.	Robot_y	-840
	9.	Goalie_x	-3130
	10.	Goalie_y	59
	11.	Post_left_seen	1 (yes)
	12.	Post_right_seen	1 (yes)
	Abstract Solution_2 Level Two	Abstract_solution2	Goal-Score
Level three	15.	Case_Index	1
	16.	Abstract_solution2	Goal-Score
	17.	Ball_seen	0 (no)
	18.	Behind_ball	1 (yes)
	19.	Robot_x	1432
	20.	Robto_y	-840
21.	Ball_relative_x	0 (NAO behind ball)	
22.	Ball_relative_y	0 (NAO behind ball)	
	Abstract Solution_3 Level Three	Abstract_solution3	Kick_to_goal,
Level four	21.	Case_Index	1
	22.	Abstract_solution3	Kick to goal
	23.	Robot_x	1432
	24.	Robot_y	-840
	25.	Goalie_x	-3130
	26.	Goalie_y	59
	27.	Angel_to_ball	0 (NAO behind ball)
	Adapted Behaviour Solution	Adapted Solution	
		Kick_far_left	-
		Kick_left	Kick angle = 70

FIGURE 4. Sample of a Concrete Case Representation and its Abstract Case

Case adaptation is performed at each level by using adaptation rules [11]. In our system, we extract adaptation rules by making analyses to the RoboCup domain. An informal example of one of our adaptation rules is:

IF the feature robot-x in the new sub-case is 1432 and the similar feature robot-x in the retrieved sub-case is 1200 and the feature robot-y in the new sub-case is -840 and the similar feature robot-y in the retrieved sub-case is 20 and the feature goalie-x in the new sub-case is -3130 and the similar feature goalie-x in the retrieved sub-case is

<u>At Level One do : Which NAO Role ?</u>		
1. Input features of the real-time	New	Role_sub_case .
2. Retrieve most similar	Retrieved	Role_sub_case.
3. Adapt the solution of the	Retrieved	Role_Abstract_Solution.
4. Output the solution	New	Role_Abstract_Solution .
<u>At Level Two do: Which NAO Skill ?</u>		
5. Append the feature of the	New	Role_Abstract_Solution
to the features of the	New	Skill_sub_case .
6. Input features of the real-time	New	Skill_sub_case .
7. Retrieve most similar	Retrieved	Skill_sub_case.
8. Adapt the solution of the	Retrieved	Skill_Abstract_Solution.
9. Output the solution	New	Skill_Abstract_Solution .
<u>At Level Three do: Which NAO Behavior ?</u>		
10. Append the feature of the	New	Skill_Abstract_Solution
to the features of the	New	Behavior_sub_case .
11. Input features of the real-time	New	Behavior_sub_case .
12. Retrieve most similar	Retrieved	Behavior_sub_case.
13. Adapt the solution of the	Retrieved	Behavior_Abstract_Solution.
14. Output the solution	New	Behavior_Abstract_Solution .
<u>At Level Four do: Which NAO walk speed and kick angel .etc?</u>		
15. Append the feature of the	New	Behavior_Abstract_Solution
to the features of the	New	Adapted_sub_case .
16. Input features of the real-time	New	Adapted_sub_case .
17. Retrieve most similar	Retrieved	Adapted_sub_case.
18. Adapt the solution of the	Retrieved	Adapted_Abstract_Solution.
19. Output the solution	New	Adapted_Abstract_Solution .

FIGURE 5. The Algorithm of the HCBR Behavior Control

-2090 and the feature *goalie-y* in the new sub-case is 59 and the similar feature *goalie-y* in the retrieved sub-case is 120 and the behavior solution in the retrieved sub-case is *kick-right*. **THEN** the behavior solution of the new sub-case is *kick-far-left*.

3.2.4. The Algorithm of the HCBR Behavior Control. This section presents the main algorithm of our HCBR Behavior Control. As shown in Figure 5, it consists of 19 steps, which are classified into four levels. Each level follows a simple case-based reasoning cycle [7] to retrieve a similar sub-case and to adapt its solution until the final solution of the complete case is found at level four.

4. Sample Run

This section shows a sample run of our experiments. They are divided into two main sets, set (A) for Goal-Score behavior and set (B) for Dribble behavior. Figure 6

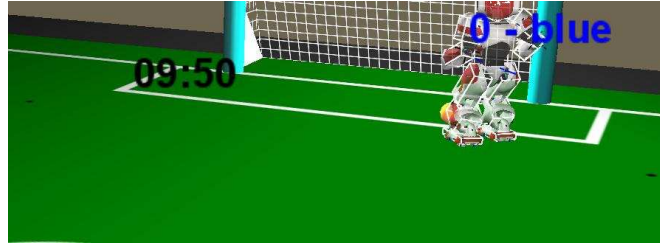


FIGURE 6. Sample Goal-Score behavior at left line of attack

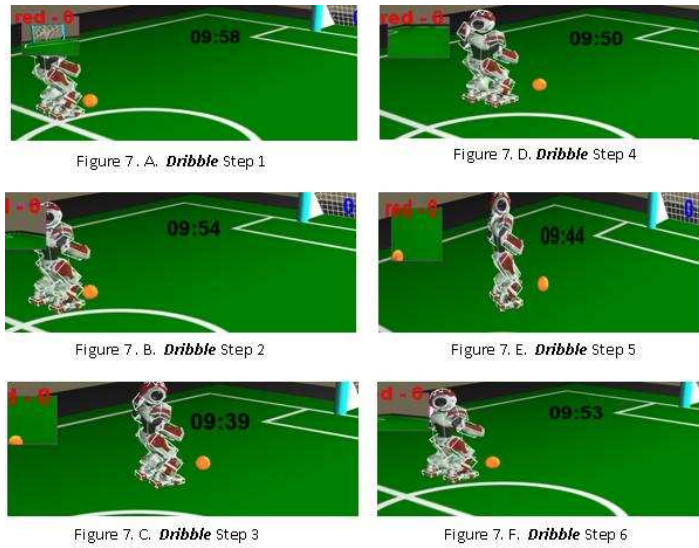


FIGURE 7. Sample Dribble Behavior at right-side of Dribble area

shows a sample run of Goal-Score behavior at line of attack. Figure 7 shows a sample run of Dribble behavior.

5. Experiment Results

In our experiments, we use 120 cases stored in the case-memory and 80 cases are used for testing the performance accuracy. All the cases are used to control the behaviors of Attacker role of NAO. These cases are further classified into two groups, which are the Goal-Score and the Dribble cases. Table 1, shows the performance accuracy of our hierarchical CBR behavior control. As shown, the accuracy rate at Level-One is very high almost 100 % because all our cases in the case-memory are only attacker cases and we do not have goalie cases. Also, high accuracy at Level-Two reaches 93.3 % and at Level-Three 80.8 %, this is due to the following main factors:

i. Usage of the GPS sensor of the Webots Simulation System to localize the robot. This gives accurate real-time positioning of NAO robot. ii. These position values are divided into ranges to simplify the retrieval process. iii. A fixed set of adaptation rules are used specifically for each set of test cases.

TABLE 1. Performance Accuracy of the Case-Based Behavior Control

phase	no.test cases	test cases classified	accuracy rate	no.of adaptation rules
Level one	120	120	100 %	4
Level two	120	112	93.3 %	12
Level three	120	97	80.8 %	35
Level four	120	86	71.66 %	63

However, the accuracy rate at Level-Four is low because this is the lower level of our HCBR Behavior Control and it needs a huge amount of adaptation rules to cover all possible numeric values of features. This is not feasible [11] and a learning mechanism should be added to learn adaptation rules [13].

6. Conclusion and Future Work

This study illustrates a new adaptive behavior control for humanoid soccer robot. It is based on case-based reasoning technique. The main aim of this research is to develop an adaptive behavior control to make our NAO robot fully autonomous and adapt its soccer behaviors to dynamic soccer game. The controller has a hierarchical scheme, which simulates the execution process of the RoboCup Soccer behaviors. It also enables NAO to learn from its experience and add new experience into its case-memory. The decomposition of features into a hierarchy of levels helps to reduce the complexity of overall behavior execution in real-time. A high performance is achieved at all the three levels and a relatively good performance rate is achieved at the fourth level. In future work, the controller will be tested using rest of soccer behaviors for goalie and defender players. Also, neural networks will be combined with case-based reasoning to learn adaptation rules to reduce the complexity of hand-coded rules and improve the overall controller accuracy. Also, other similarity formulas will be tested like fuzzy similarity measures to improve case retrieval.

References

- [1] Humanoid Team Humboldt , <http://www.naoteamhumboldt.de/index.php>
- [2] <http://world.honda.com/ASIMO>
- [3] RoboCupSoccer Humanoid League , <http://www.HumanoidSoccer.org/>
- [4] Artificial Intelligence, <http://www.ki.informatik.hu-berlin.de/>
- [5] Kolodner, J. : Case-Based Reasoning. Morgan Kaufmann Inc., San Francisco (1999).
- [6] Katic, D., Vukobratovic, M. : Survey of Intelligent Control Techniques for Humanoid Robots. *J. Intelligent and Robotic systems*. 37, 117-141 (2003).
- [7] Hoffman, F.: An overview on Soft Computing in Behavior Based Robotics. In: the 10th International Fuzzy Systems Association World Congress , IFSA 2003 , Fuzzy Sets and Systems , pp. 544-551, Istanbul, Turkey, (2003).
- [8] Ros, R. , Lpez de Mntaras, R. , Sierra, C.; Arcos, J. : A CBR System for Autonomous Robot Navigation. In *Proceedings of the Catalan Conference on Artificial Intelligence ,CCIA-05* , Lpez, B. et al. eds., *Frontiers in Artificial Intelligence and Applications*, N. 131, pp. 299 - 306. IOS Press, (2005).
- [9] Urdiales, J., Vzquez-Salceda, E.J. , Perez, Sandoval, F.: A CBR Based Pure Reactive Layer for Autonomous Robot Navigation. In: 7th IASTED International Conference on Artificial Intelligence and Soft Computing, Banff, Canada, pp. 99-104 (2003).
- [10] Espinosa, R.R., Veloso, M.: Executing Multi-Robot Cases Through a Single Coordinator. In: 6th AAMAS'07 International Joint Conference on Autonomous Agents and Multi-Agent Systems, Honolulu, Hawaii (2007).

- [11] El Bagoury, B.M.,Burkhard,H-D., Salem, A.B.M. : Design of Case-Based Behaviour Control Algorithm for Humanoid Soccer Robot. In Proceedings of Concurrency, Specification and Programming of Conference, CSP'2008, Berlin, Germany (2008).
- [12] Aldebran NAO Robot , <http://www.aldebran-robotics.com>
- [13] Burkhard, H.D., Salem, A.B., El-Bagoury,B.M.: Case-Based Motion Controller for Humanoid Robot. In: AIPR-08 International Conference of Artificial Intelligence and Pattern Recognition,pp.32-37, Florida, USA, (2008)

(Bassant Mohamed El-Bagoury) FACULTY OF COMPUTER AND INFORMATION SCIENCE, AIN-SHAMS UNIVERSITY, CAIRO, EGYPT

E-mail address: {bassant, hdb}@informatik.hu-berlin.de

(Abdel-Badeeh M. Salem) FACULTY OF COMPUTER AND INFORMATION SCIENCE, AIN-SHAMS UNIVERSITY, CAIRO, EGYPT

E-mail address: abmsalem@absalem@asunet.shams.eun.eg

(Hans-Dieter Burkhard) INSTITUTE OF INFORMATICS, HUMBOLDT UNIVERSITY IN BERLIN, GERMANY

E-mail address: {bassantai, abmsalem}@yahoo.com