

# Enhancing Case-Based Retrieval Engine with Case Retrieval Nets for Humanoid Robot Motion Controller

Meteb M. Altaf, Bassant M. El Bagoury, Fahad Alraddady, and Said Ghoniemy

**Abstract**—An efficient retrieval of a relatively small number of relevant cases from a huge case base is a crucial subtask of Case-Based Reasoning. Moreover, Motion Controlling for Humanoid Robot is a very complex problem. In this paper, we propose the application of case-retrieval nets techniques in the design of our previously proposed motion controller model for humanoid robots. It depends on case-based reasoning (CBR) methodology. Our main goal is to enhance the retrieval accuracy of the case-based controller of the humanoid soccer. The controller is being implemented in the framework of Webots Simulation Tool for the NAO Humanoid Robot. The main motivation of this paper is to improve the retrieval accuracy of our HCBR behavior controller, develop an automatic real-time CBR-Retrieval Algorithm for robot, and improve the storage capacity of the case-memory. We also describe the implementation of our extended retrieval CBR algorithm that shows good results for controlling the NAO. Future research directions and ideas for developing each module are also discussed.

**Index Terms**—Humanoid robot, RoboCup, artificial intelligence, case-based reasoning, webots, motion controller.

## I. INTRODUCTION

The RoboCup [1], [2] competition is a long-term goal of winning against the FIFA world champion [3], [4]. Our Humanoid Team of Humboldt University [3]-[5] participated in 2006 for the first time in humanoid league. Motion planning for humanoid robots suffers from many problems. The first problem is the high dimensionality of the configuration space. The second is the need to satisfy dynamic and static constraints for motion stability. The third is the need to navigate in dynamic environment as soccer. Moreover, building motion controllers on real robots increases the overall complexity. Hardware easily gets broken and experiments need manual supervision.

Case-Based Reasoning (CBR) is a reasoning methodology that simulates human reasoning by using past experiences to solve new problems [6]. The most crucial CBR tasks involve case indexing, representation, retrieval and adaptation.

Manuscript received October 9, 2014; revised December 24, 2014. This work was supported in part by Director of Center of Robotics and Intelligent Systems, Dr. Meteb M. Altaf, at the King Abdel Aziz City for Science and Technology (KACST), Riyadh, Kingdom of Saudi Arabia.

M. Altaf is with Center of Robotics and Intelligent Systems at King Abdel-Aziz City for Science and Technology, KACST, Riyadh, Kingdom of Saudi Arabia (e-mail: maltaf@kacst.edu.sa).

B. Elbagoury was with Humboldt German Team of Robotics, NAO Humanoid Team, Berlin, Germany and Robotics and Computer Science at Faculty of Computers and Information Sciences, Ain Shams University, Cairo, Egypt (e-mail: bassantai@yahoo.com).

S. Ghoniemy is with Faculty of Computers and Information Science, Ain Sham University, Cairo, Egypt. He is also with the College of Computers and Information Technology, Taif university, Taif, KSA (e-mail: ghoniemy@gmail.com).

Our Humanoid Team Humboldt builds a simulated tool called Simloid for the Bioloid [7], [8]. Currently, we are also using the Webots Simulation Tool for Nao Robot [9]. Our main goal is to develop a motion controller for fully autonomous humanoid robot to navigate in unstructured environment.

In this paper, we propose a new case-based motion controller design for humanoid robots. It is currently being implemented in the framework of Webots [9]. The model design and description of each module are presented. We also present the results of our first Case-based algorithm for poses and basic walk control. The paper is organized as follows; Section II describes CBR for robotics, Section III describes robotics platform, Section IV describes the proposed CBR motion controller, Section V introduces the CBR algorithm along with the new proposed retrieval algorithm. Finally Section VI is experimental results.

## II. CBR FOR ROBOTICS

In robotics, CBR has been applied recently for many robotics types and tasks. For example, Hongwei [10] uses CBR to evolve robust control programs for humanoid robots. Arcos *et al.*, [11] use CBR for autonomous mobile robot navigation. Kruusmaa [12] uses CBR for navigation but in uncertain environment, and Urdiales [13] presents a new reactive layer that uses CBR for robot navigation. CBR has also been widely applied in the RoboCup domain; Raquel *et al.* [14] use CBR to define coordination of behaviors for multi-robots. Also, they use CBR for retrieving and reusing old game plays for robot soccer [15]. Timo [16] uses CBR for opponent modeling in multi-agent systems. Ahmadi *et al.* [17] use CBR for prediction of opponents movements in multi-agent robotic soccer. Karol *et al.* [18] use CBR for high-level planning strategies for robots playing in the Four-Legged RoboCup.

## III. ROBOT PLATFORM FOR NAO-TEAM HUMBOLDT

The NAO-Team Humboldt was founded at the end of 2006 and consists of students and researchers from the Humboldt-University in Berlin. Some of the team members have had a long tradition within RoboCup by working for the Four Legged league as a part of the German Team in recent years. Though we used some concepts and ideas from the GT-platform, the software architecture was written totally new. Additionally, we developed several tools such as Robot-Control and MotionEditor for testing, debugging or for creating new motion nets.

The first NAOs as shown in Fig. 1, arrived in May 2008, so we had only 2 months for developing and testing algorithms

before we participated in RoboCup 2008 in Suzhou. Despite this fact, we achieved the 4<sup>th</sup> place.

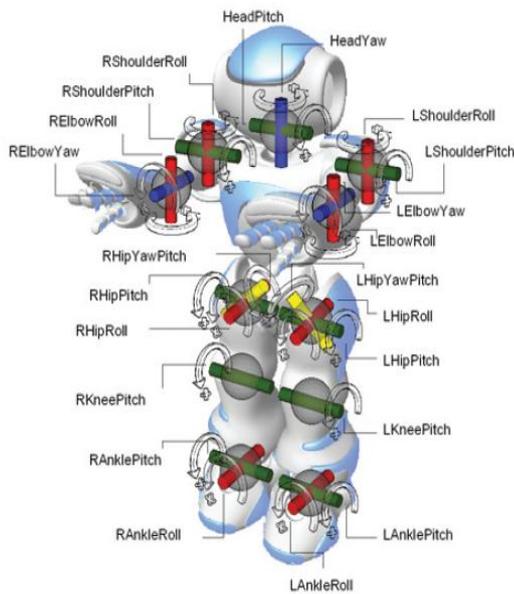


Fig. 1. Nao humanoid robot.

### A. Vision System

Our vision system worked on YUV images with a resolution of  $160 \times 120$ . Because of the limitations of such a small resolution, as finding small objects and accuracy of recognition, we will increase resolution to VGA 2015. Most of the algorithms are based on colour classification. The classification is done using two different means - a look up table and especially for complementary colours, linear colour space segmentation. Since it is, even in this resolution, inefficient to classify the whole picture, as shown in Fig. 2, we employ a grid for this task. The grid is laid over the picture, has a resolution of  $80 \times 60$  and thus classifies only one in four pixels. It provides a -list of all pixels from a given color class to the subsequent procedures which are as follows.

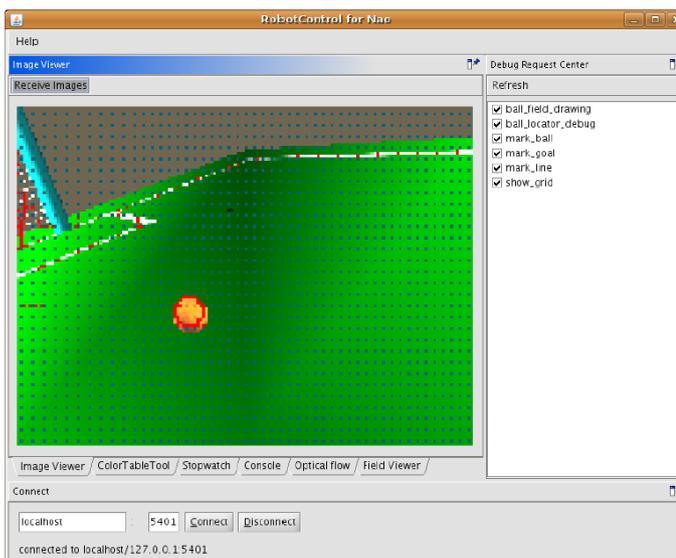


Fig. 2. Robot Image viewer with some enabled debugs requests.

### B. Goal Detection

The algorithm for goal detection relies mainly on the

calculation of statistical measures for lists of equally coloured pixels provided by the grid. For one class of goal coloured pixels the algorithm first looks for the best 2 candidate pixels maximizing  $x^2+y^2$  or  $x^2-y^2$  (thus minimizing the distance to the lower left/right corner) while in the same iteration calculating all image moments up to 2<sup>nd</sup> degree of all pixels from this colour class. Those moments are used to calculate the main axis for the distribution of pixels having goal colour. The axis orientation gives a good hint for both goal posts as shown in Fig. 3a), on the image having gathered this information the algorithm knows how to interpret the 2 candidates found before and starts to explore via region growing from every pixel representing a goal posts base point thus further defining the candidates and the corresponding percept.

### C. Line Detection

Line Detection is done without color classification. It instead scans the whole picture in horizontal and vertical directions looking for rises and subsequent falls in the images gray value function indicating the possible start and end of a line. Having found two of those points the edge angle is calculated by the Sobel operator and then averaging the two pixels positions resulting in a new point and corresponding angle in the middle of these points. Points found this way are then first clustered by their angle and second by their probability of lying on one line. Clusters with a sufficient number of members are then accepted as lines as shown in Fig. 3a).

### D. Robot Detection

Robot Detection as shown in Fig. 3b), is done using red or blue colour areas in the image. Those blobs representing distinct parts of a robot (i.e. head, shoulders, feet etc.) must have certain attributes such as certain area, centre of mass or orientation. Body part candidates identified this way are then tried to be grouped to form a robot. The position and orientation of the detected robot can easily be extracted from the geometric relations between the constituting colour areas.

### E. World Model

Our approach to represent the world state is to use different models for different objects. In having this separation we can have very effective and special models for each individual object type. We distinguish two modelling approaches – self localization, ball modelling, and player modelling.

### F. Ball Modeling

Tracking the ball is most important for the attacker and for the goal keeper. Since playing passes was hard with the NAOs so far, which would make global positions necessary; we use a local model for each robot to track the ball. Still, by using self localization information we were able to communicate global ball we also want to have a player model to recognize friendly passing partners and to avoid kicking the ball into opponent players. Right now we want to find out which kind of model fits our needs best. Therefore we evaluate the advantages of a model which tracks all the different players separately against those of a model which just takes occupied regions into account.

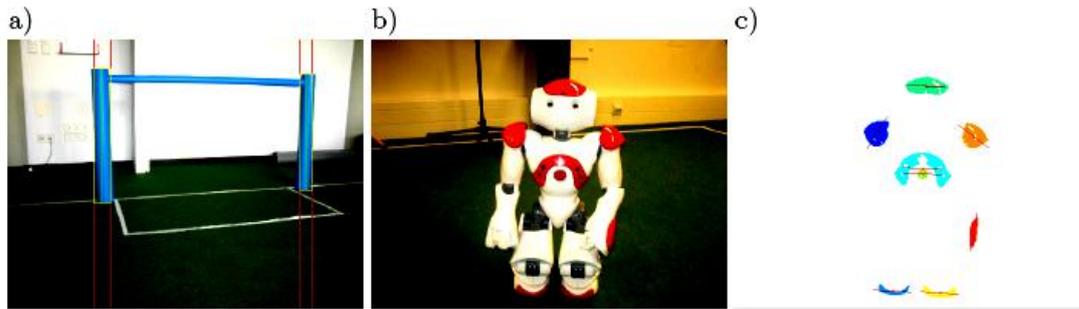


Fig. 3. a) A recognized goal, detected by two rectangles framing the goal posts b) a seen robot c) detected robot areas including the excentricity.

### G. Robot Detection

Robot Detection is done using red or blue colour areas in the image. Those blobs representing distinct parts of a robot (i.e. head, shoulders, feet etc.) must have certain attributes such as certain area, centre of mass or orientation. Body part candidates identified this way are then tried to be grouped to form a robot. The position and orientation of the detected robot can easily be extracted from the geometric relations between the constituting colour areas as shown in Fig. 3c).

### H. World Model

Our approach to represent the world state is to use different models for different objects. In having this separation we can have very effective and special models for each individual object type. We distinguish two modelling approaches – self localization, ball modelling, and player modelling.

#### I. Ball Modeling

Tracking the ball is most important for the attacker and for the goal keeper. Since playing passes was hard with the NAOs so far, which would make global positions necessary; we use a local model for each robot to track the ball. Still, by using self localization information we were able to communicate global ball

#### J. Player Modeling

Having data over recognized field players we also want to have a player model to recognize friendly passing partners and to avoid kicking the ball into opponent players. Right now we want to find out which kind of model fits our needs best. Therefore we evaluate the advantages of a model which tracks all the different players separately against those of a model which just takes occupied regions into account.

## IV. CASE-BASED MOTION CONTROLLER FOR NAO HUMANOID ROBOT

The goal of our Humanoid-Team-Humboldt [5] is to develop a motion controller for fully autonomous humanoid robot to navigate in unstructured environment.

This section describes our new proposed case-based motion controller for humanoid robots. Its main architecture is shown in Fig. 4, it consists of four main Modules: CBR biomechanical Module, CBR navigational Module, case-based keyframe motion planning Module and CBR gait balance Module. Also, it has three CaseBases: CaseBase of

keyframes of body poses, sub-CaseBase of keyframes of balanced body poses and a CaseBase of keyframes of environment paths. The next sub-sections define the task of each Module.

#### A. CBR Biomechanical Module

This is a Case-Based Medical Expert System. Its task is to collect experience of the biomechanical expert doctors and built a CaseBase of human-like body poses. We believe that this will help in developing a motion controller that mimics real human motions.

Development directions will include: knowledge engineering phase of the biomechanical domain [18], using fuzzy retrieval algorithms to retrieve similar cases, using and modifying our developed adaptation model [19] to adapt the unusual body poses. The architecture of this Module will simply follow the CBR cycle. This Module will generate two CaseBases, which are: CaseBase of keyframes of all biomechanical body poses, and Sub-CaseBase of keyframes of balanced biomechanical body poses. A keyframe here is defined as a case of body poses of all joint angles.

#### B. CBR Navigational Module

This is a case-based system for robot navigation. Its task is to use CBR to navigate in unstructured environment. Research directions in this Module will include: enhancement of retrieval algorithms for case-based navigation using fuzzy logic and development of new adaptation algorithms for case-based navigation and obstacle avoidance in unstructured environments. These Research directions in this Module will focus mainly on developing an independent retrieval-adaptation model for robot motion planning in unstructured environments. One of the tasks of the adaptation model is to function as the transitions in the keyframe-transition structure [8], [9]. A keyframe is a structure that keeps all joint angles of the robot current pose. A transition is a decision to transit from one keyframe to another. Poses and motions are executed by transitions between keyframes.

#### C. Case-Based Keyframe Motion Planning Module

This is the main Module of the case-based motion controller. Its task is to plan for the next body pose and generate the next walking pattern that the humanoid robots should adopt. As shown in Fig. 4 it works as follows: first the current keyframe is seen as an input. Then the most similar keyframe is retrieved from the casebase of body poses. The

retrieval algorithm also takes into account the current path navigated by the CBR navigational module. Finally, the retrieved keyframe is adapted according to the current navigated path. This adapted keyframe is proposed as the next body pose.

**D. CBR Gait Balance Module**

This is a case-based fault diagnosis expert system. Its task

is to test and monitor the balance state of the generated walking pattern. It will simply follow the CBR cycle and will use the Sub-CaseBase of balanced keyframes. In case of not accepted walking pattern, it will return back to the Case-Based Keyframe Motion Planning Module to re-generate another walking pattern. This feedback cycle will continue until a suitable walking pattern is generated.

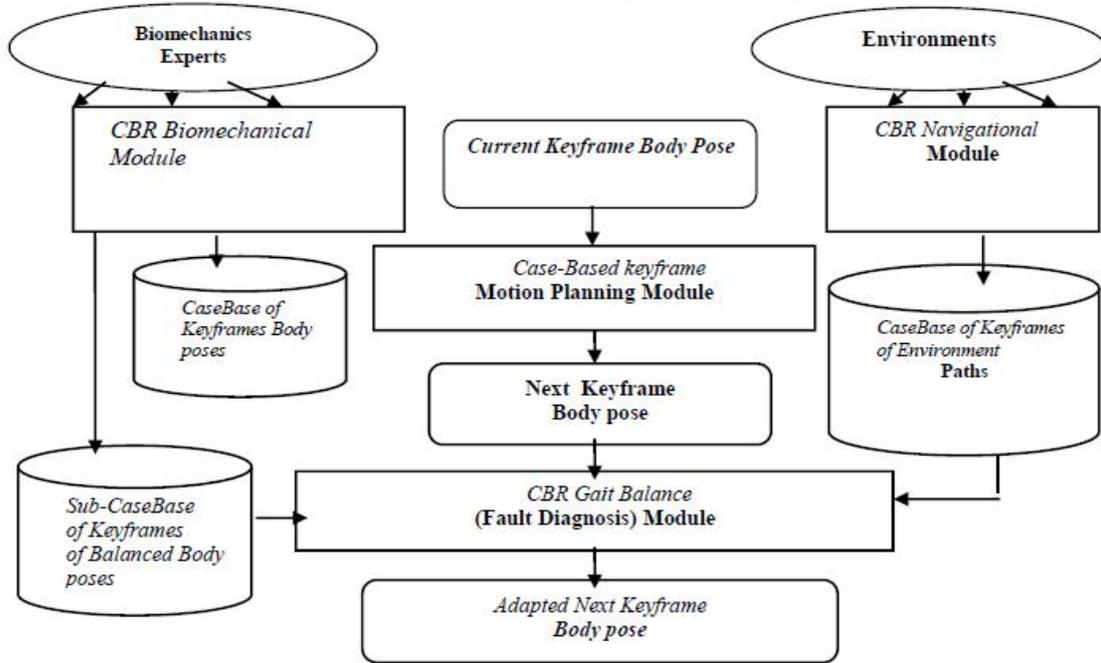


Fig. 4. Case-Based Keyframe motion controller for humanoid robots.

**V. REAL-TIME CASE-BASED ALGORITHM FOR KEYFRAMES GENERATION**

We are currently implementing our new algorithms for our case-based motion controller. In this section, we describe our first Case-based algorithm for Nao Robot keyframes generation. Its architecture is shown in Fig. 5. It consists of three main modules, which are case input, retrieval and adaptation. It also includes a case-memory of cases and a rule-base of adaptation rules.

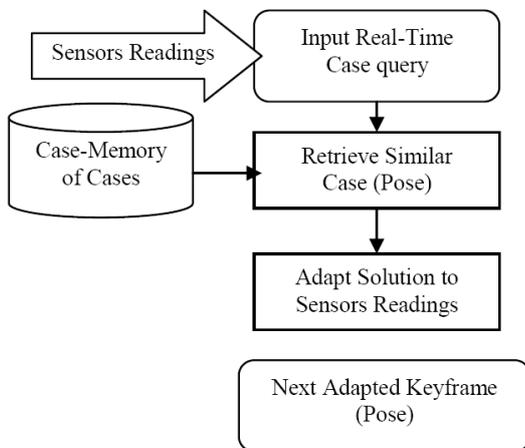


Fig. 5. Architecture of real-time case-based Keyframe algorithm.

**A. The Case-Memory**

NAO humanoid robot has ten sensors, one distance sensor

for measuring distance, eight force sensors, four in each foot and one accelerometer sensor for measuring acceleration. In addition, Nao Robot has twenty two Joints. Taking these in consideration, therefore each case in our case-memory consists of thirty two features and it is represented as a frame [9]. One sample case of our cases is shown in Fig. 6. The case is decomposed into two parts:

Case <problem, solution>. Case problem consists of the ten sensors features, while case solution consists of the twenty-two joints features.

	Case Problem Features	Case query sample
Distance Sensor	UsLR	4 m
	USRR	0 m
Force Touch sensors (four for each leg)	RFsrBR	40
	RFsrBL	40
	RFsrFL	60
	RFsrFR	60
	LFsrBR	40
	LFsrBL	40
	LFsrFL	60
	LFsrFR	60
Accelerometers (for x, y and z axis)	Accx	0
	Accy	0
	Accz	30

	Case Solution Features	Case Solution(Pose) (Keyframe) sample
Right Leg	RHipYawPitch	10
	RHipPitch	-25
	RHipRoll	-20
	RkneePitch	0
	RAnklePitch	-20
	RAnkleRoll	-20
Left Leg	LHipYawPitch	10
	LHipPitch	-20
	LHipRoll	0
	LkneePitch	30
	LAnklePitch	-80
	LAnkleRoll	-40
Right arm	RShoulderRoll	80
	RShoulderPitch	50
	RElbowRoll	60
	RElbowYaw	40
Left arm	LShoulderRoll	80
	LShoulderPitch	50
	LElbowRoll	60
	LElbowYaw	40
Head	HeadYaw	-120
	HeadPitch	-45

Fig. 6. A sample of the Nao robot case frame.

### B. Case Retrieval Nets

This section presents the main algorithm of our CRN-HCBR behavior control. As shown in Fig. 6, it consists of 12 steps, which are classified into three levels. Each level uses a CRN [2] to retrieve a similar sub-case and apply propagation adaptation rules to adapt its solution until the final solution of the complete case is found at level three.

#### CRN-HCBR Algorithm:

- i) **Input** Real-Time IE's Case Query of Level 1
- ii) **Retrieve** similar IE's using CRN
- iii) **Adaptation Propagation Rules** to adapt Abstract case nodes of Robot Role.
- iv) **Output1:** Adapted Role solution
- v) **Backward Reasoning** Adapted Role & Append as new IE to Level 2.
- vi) **Retrieve similar IE's** using CRN.
- vii) **Adaptation Propagation Rules** to Abstract case nodes of Robot Skills.
- viii) **Output2:** Adapted Skills solution
- ix) **Backward Adapted Skills** & Append as new IE to Level 3.
- x) **Retrieve similar IE's** using CRN.
- xi) **Propagate** to Abstract case nodes of robot behaviors.
- xii) **Apply Adaptation function** of case-based NAO behaviors.

The main steps of CRN-HCBR are described as follows:

**Step 1: Input Case-Query.** This is done by real-time sensor IE's readings from robot simulation environment.

**Step 2: Retrieve Similar IE's.** Retrieve module uses only local similarity functions to retrieve similar IE's. Two local similarity functions are used, a Boolean function that is used to compute local similarity of Boolean features and it is defined as:

$$Sim(Ni, Ri) = |1 - (Ni + Ri)|$$

Another local similarity function to compute similarity between features of real values as *Robot\_x* and *Robot\_y* features. This function is defined by Burkhard [6]:

$$Sim(Ni, Ri) = 1 / \{1 + |Ni - Ri|\}$$

**Step 3: Apply Adaptation Propagation rules.** In this algorithm, adaptation rules, which will be defined also in coming section, are used to propagate to case nodes and propose solutions.

**Step 4: Output 1.** Find adapted role solution. This is the first solution that results for robot role as attacker or Goalie.

**Step 5: Backward Reasoning.** This is to append Robot Role solution to case query IE's in real-time RoboCup soccer domain. This updates the Case query IE's and thus activates new IE's query for the second level automatically.

**Step 6: Retrieve Similar IE's.** This step also applies the same local similarity functions but for Level two. This is to find solution of Robot Skill as Goal-score, Dribble or pass.

**From Step 7 to Step 11.** The previous steps are repeated recursively from Step 7 to Step 11 until the lower level primitive behaviors are executed for the robot.

### C. Adaptation and Keyframe Generation

The main task of our case-based algorithm is to take decisions on the behavior level [10], [13], [17]. This means to decide which keyframe to execute next. These decisions are done by using a set of adaptation rules, which first check for similarity conditions and then take decisions. These decisions can be a single generated keyframe as fall-down or a sequence of generated keyframes as walk-forward.

An example of one of our adaptation rules is:

**IF** (Similarity  $\geq$  90%)  
**AND** Sensor\_Readings in range of Stand\_up  
**THEN** Take decision\_1 (generate keyframe) of Walk\_Forward

As shown in this example, our adaptation rule consists of three parts: **IF ... AND ... THEN**. In the **IF part**, we test for the global similarity value. In the **AND part**, we test again for the sensor readings of the case query to determine the current position of the Robot in order to generate the suitable keyframe(s) in the **THEN part**.

## VI. EXPERIMENTAL EVALUATION

In this section, the experimental results of the CRN-HCBR algorithm are discussed. Our experiments are done in the simulation environment of *Webots* integrated with Visual Studio for programming our CBR behavior control algorithm. These are done in the framework of the project of NAO Humanoid Team Humboldt [5]. We conduct four main experiments. Experiment (A) is for testing the retrieval accuracy of the CRN. Experiments (B) is for testing the overall performance measure of the CRN-HCBR algorithm for Attacker robot. The measures used for testing retrieval accuracy of the CRN for cases retrieval are:

**Number of IE's and Case nodes in the Case-Memory:** The main goal of CRN should to reduce the size of case-memory. The size of CNR memory is measured by number of IE's and number of case nodes but this must be

independent of the overall total number of cases stored in case-memory.

**Retrieval Accuracy.** It means that the required set of case nodes are retrieved correctly during the execution of the retrieved process.

**Retrieval Efficiency.** It means the ability of our CRN retrieval to give the same set of retrieved case nodes at different similarity ranges, for similar test query cases. That is to access relevant cases. Access of these cases should avoid exhaustive search in memory.

**Completeness:** assures that every sufficiently similar case in memory will be found during retrieval.

**Experiment: Testing the accuracy of CRN for Attacker Robot**

In this experiment, we test the accuracy of the CRN in case of robot role acts as Attacker in RoboCup simulated soccer. The accuracy of is measured for the two CRN's used at level two and level three of our CRN-HCBR. It is not measured at level one because we already fix the robot role to be Attacker.

**Number of IE's and Case nodes in the Case-memory at level two:** the main task of level two is to determine the skill of the Attacker robot.

**Number of IE's and case nodes in Case-memory:** The skills to be decided are Goal-Score or Dribble. We define a formula to calculate the number of IE's stored in CNR case-memory as:

$$IE\_CRN = N \times D = 14 \times 10 = 140 \text{ IE's.}$$

where,

IE\_CRN: The number of IE's stored in case-memory of the CRN

N: the real number of features defined from RoboCup soccer domain

D: the number of associated features results from discretization of real features. We fix these ranges to be equal nearly 10.

This proves a big reduction in the size of the case-memory. The number of stored case nodes is 650, these represents the real number of total cases for Attacker robot. Therefore, instead of storing 650 IE's, it is reduced to 140 IE's. This is illustrated in Table I.

We also define a simple formula for defining the number of case nodes included in CRN. Our formula is:

$$\text{CaseNodes\_CRN} = \frac{\text{Total Number of Cases in flat Case Memory}}{\text{Total Number of Cases in flat Case Memory}}$$

where Total number of Cases is total number of complete cases stored in standard flat structure case-memory.

In our experiment, the total number of cases is 650 cases. This means the number of Case nodes is also equals 650. This is shown in Table I.

**Retrieval Accuracy.** In order to improve the retrieval accuracy of our CRN, the values of each IE are discretized into ranges. This is also done to reduce the error of sensor readings of the *Webots* simulation tool. Table I shows as a sample a complete list of Robot localization IE's used in this experiment along with their ranges. The retrieval accuracy of CRN at level is very high as it reaches 97 %. Also, the CRN at level 3 reaches 92%. The results are shown in Table I. The

retrieval accuracy is calculated by our formula given as:

$$\text{Retrieval\_Accuracy} = CC / TC$$

where,

CC: Number of sets of case nodes retrieved correctly by our algorithm

TC: Number of total sets of case nodes included in the CRN-HCBR controller.

**Retrieval Efficiency.** The retrieval efficiency of the CRN at level two was very high as it reaches 98 %, while at level three it reaches 96%. This is because the cases used for querying are divided into similar sets, which increases the overall retrieval efficiency. The results are shown in Table I Our formula for retrieval efficiency is defined as:

$$\text{Retreival\_efficiency} = SC / TQ$$

where,

SC: Number of sets of case nodes retrieved correctly during cases query runs.

TQ: Total number of sets of case queries used for testing.

**Completeness.** It shows low performance at level two as it reaches 70 % and at level three it reaches only 60 %. This due to a huge number of IE's ranges need to incorporated further in the system or a complete set of Adaptation rules. These is usual is infeasible and a learning mechanism is essential in these cases.

TABLE I: PERFORMANCE ACCURACY OF CRN

	No. of IE's	Accuracy	Efficiency	Completeness
CRN Level Two	140	97 %	98 %	70 %
CRN Level Three	200	92 %	96 %	60 %

VII. CONCLUSION AND FUTURE WORK

In this paper, the design of a case-based motion controller for Humanoid robots is represented. It depends on case-based reasoning methodology. It is main goal is to handle the problems of motion planning tasks for humanoids robots. That is to generate adaptive balanced human-like walking patterns, which are able to navigate in dynamic and unstructured environments, to avoid obstacles and collision with multiple robots. This model is currently being implemented in the framework of Webots Simulation tool for the Nao Robot.

This paper also presents the first case-based algorithm that has been implemented for generating some basic keyframes. However, first experiments show success of our first case-based algorithm and its applicability for generating humanoid robot motions.

In our future work, the complete implementation of our model will be done in our Nao humanoid project. As a next step, fuzz logic will be used for case retrieval in order to retrieve more reliable cases. Also, the algorithm will cover decisions of internal keyframes of each state, as generating a

walk step to simulate real human walking cycle. This will include knowledge engineering phase of the human gait analysis and walking cycle.

#### ACKNOWLEDGMENT

Thanks to Dr. Meteb M. Altaf, Director of Robotics and Intelligent systems Center, King Abdel-Aziz City for Science and Technology, (KACST), Riyadh, Kingdom of Saudi Arabia. Special thanks to Prof. Dr. Said Ghoniemy, Professor of Computer Science at College of Computers and Information Technology at Taif University, taif, Kingdom of Saudi Arabia.

#### REFERENCES

[1] Humanoid League Competition Rules. (2006). [Online]. Available: <http://www.humanoidsoccer.org/rules.html>

[2] J. Baltes and J. Anderson, "Humanoid robots: Hiro and DaiGuard-RS," in *Proc. RoboCup 2005 Humanoid League Team Descriptions*, Osaka, Japan, 2005.

[3] Praktikum – Moderne Methoden der KI. [Online]. Available: <http://www.ki.informatik.hu-berlin.de/lehre/ss06/mmk-prakt.html>

[4] H. D. Burkhard, D. D. Fujita, P. Lima, R. Murphy and R. Rojas, "The road to robocup 2050," *IEEE Magazine of Robotics and Automation*, vol. 9, no. 2, pp. 31-38, 2002.

[5] Humanoid Team Humboldt. [Online]. Available: <http://www.humanoidteamhumboldt.de>

[6] J. L. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann Publishers, 1993, pp. 347-366.

[7] H. D. Burkhard and H. Hein, "Simloid-evolution of biped walking using physical simulation," Diploma thesis, Humboldt University in Berlin, Institute of Informatic, 2007.

[8] H. Heind and M. Hild, "Simloid-research on biped robots controller, using physical simulation and machine learning algorithms," in *Proc. of German Workshop*, CS&P06, 2006.

[9] Webots: Robot Simulator – Overview. [Online]. Available: [www.cyberbotics.com/products/webots/](http://www.cyberbotics.com/products/webots/)

[10] H. Liu and H. Iba, "A hierarchical approach for adaptive humanoid robot control," in *Proc. CEC2004 Congress on Evolutionary Computation*, 2004.

[11] L. J. Arcos, R. L. Mantaras, C. Sierra, and R. Ros, "A CBR system for autonomous robot navigation," in *Proc. CCIA'05, Fronteries in Artificial Intelligent and Applications*, 2005, pp. 299-306.

[12] M. Kruusmaa, "Global navigation in dynamic environments using case-based reasoning," *Journal of Autonomous Robots*, vol. 14, no. 1, pp.71-91, 2003.

[13] J. Urdiales, E. J. Vázquez-Salceda, and F. Sandoval, "A CBR based pure reactive layer for autonomous robot navigation," in *Proc. 7th IASTED International Conference on Artificial Intelligence and Soft Computing*, Banff, Canada, 2003, pp. 99-104.

[14] R. R. Espinosa and M. Veloso, "Executing multi-robot cases through a single coordinator," in *Proc. the 6th AAMAS'07 International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Honolulu, Hawaii, 2007, pp. 243-250.

[15] R. Ros, M. Veloso, R. López de Mántaras, C. Sierra, and J. L. Arcos, "Retrieving and reusing game plays for robot soccer," in *Proc. ECCBR-2006*, LNAI 4106, 2006, pp. 47-61.

[16] T. Steffens, "Similarity-based opponent modelling using imperfect domain theories," in *Proc. CIG'05 IEEE 2005 Symposium on Computational Intelligence and Games*, G. Kendall and S. Lucas, Eds., 2005, pp. 285-291.

[17] M. Ahmadi, A. K. Lamjiri, M. M. Nevisi, J. Habibi, and K. Badie, "Using two-layered case-based reasoning for prediction in soccer coach," in *Proc. the International Conference of Machine Learning, Models, Technologies and Applications*, 2004.

[18] A. Karol, B. Nebel, C. Stanton, and M. Williams, "Case based game play in the RoboCup four-legged league part I: The theoretical model," in *Proc. RoboCup*, 2003, vol. 3020 of LNCS, pp. 739-747.

[19] A. B. M. Salem and B. El Bagoury, "A case-based adaptation model for thyroid cancer diagnosis using neural networks," in *Proc. FLAIRS Conference FLAIRS2003*, USA, Florida, 2003, pp. 155-159.



**Meteb M. Altaf** was born in Makkah, Saudi Arabia. He got his BSc. degree in mechanical engineering from Umm Al-Qura University. At the School of Engineering and Design, Brunel University, London, UK, Dr. Altaf started his postgraduate studies. He got his PhD degree in biomedical/mechanical engineering in 2009. Since then, he joined King Abdulaziz City for Science and Technology in Riyadh, Saudi Arabia, as an assistant research professor of robotics and mechanical engineering.

Now Dr. Altaf is an assistant director of the National Centre of Robotics and Intelligent Systems for Scientific Affairs. He is also teaching couple of courses as a part timer professor in King Saud University, Riyadh, Saudi Arabia. His research areas of interest include robotics: medical robotics, rehabilitation robotics, security robotics, educational robot, industrial robotics, manipulator robotics; biomedical engineering: bio-fluid mechanics, biomaterials, biomechanics; mechanical engineering: fluid mechanics, heat transfer, mechatronics; and engineering management: team leading, management skills, project management.