A HYBRID CBR-NEURAL ADAPTATION ALGORITHM FOR HUMANOID ROBOT CONTROL BASED ON KALMAN BALL TRACKING

BASSANT MOHAMED ELBAGOURY[1], ABDEL-BADEEH M. SALEM

Abstract. Controlling autonomous, humanoid robots in a dynamic, continuous, and real-time environment is a complex task, especially behavior control and robot ball tracking problems. This paper presents a hybrid Probabilistic CBR-Neural behavior controller for NAO Humanoid soccer. It extends our previously proposed Case-based behavior controller with neural network and Kalmn filter (EKF) for ball tracking in the 3D RoboCup soccer simulation scenario [11]. This is to solve the adaptation problem in CBR and to let the robot learn the Goal-scoring behavior cases that should be executed in real-time. The learned behavior depends highly on the ball tracking results, which is shown using different experiments presented. Our experiments are conducted on the Goal-Score behavior for adapting actions of an attacker humanoid robot. The integration of neural network module for case adaptation shows a very high performance accuracy that reaches average 92.3% by integrating the ball tracking module with Kalman filter. The algorithm modules, results and future research direction are discussed in this paper.

Keywords: case-based reasoning, behaviour control, neural network, extended Kalman filter, ball tacking, RoboCup soccer, case adaptation.

1. INTRODUCTION

Controlling an autonomous, humanoid robot in a dynamic, continuous environment is a difficult task. Manually programming complex behaviors can be very time consuming and tedious, since the decisions made by the agents depend on many features and constraints imposed by the environment. Case-Based Reasoning (CBR) [5] as a paradigm for building intelligent computer systems has been applied to robot tasks such as navigation [8, 9] and behavior control [7, 11]. For example, Raquel uses CBR for action selection in cooperative robotics soccer. Berger [7] exploits past experience case-based decision support for soccer agents. Arcos *et al*., [8] uses CBR for autonomous mobile robot navigation. CBR has also been widely applied in RoboCup domain; Raquel *el al.* [14] uses CBR to define coordination of behaviors of multi-robots. Ahmadi *et al.* [15] uses CBR for

[1] Faculty of Computer and Information Sciences Ain Shams University, Cairo, Egypt, E-mail: abmsalem@yahoo.com, bassantai@yahoo.com

Rom. J. Techn. Sci. – Appl. Mechanics, Vol. 57, N^{os} 2–3, P. 206–216, Bucharest, 2012

prediction of opponent's movements in multi-agent robotic soccer. Celeberto, L.A. [6] uses CBR for high level planning strategies for robots playing in the Four-Legged RoboCup. However, in the RoboCup [6, 7] domain the overall complexity increases especially in behaviour control. This is due to the complexities and dynamics of robot environment. Complex behaviors such as Goal-Score should be executed correctly in real-time. We have previously developed a CBR behavior control Platform for Humanoid Soccer RoboCup with NAO Team Humboldt [12]. However, still many problems are not addressed like ball localization and tracking. Moreover, adaptation in CBR engine is difficult because it needs a lot of adaptation knowledge [11].

In this paper, we extend our research for the CBR behavior controller and integrate Neural Network for Adaptation and ball tracking and localization that based on Probabilistic algorithm used in RoboCup research [14]. Our motivation in this work is to develop a more accurate controller for the Humanoid Robot that depends on extended Kalman filter (EKF) [1, 3, 4].

2. NAO HUMANOID ROBOT AND METHODOLOGIES

Humanoid Robots are a recent challenge in intelligent robotic control and autonomous agents. Humanoid Robots requires artificial intelligence (AI) techniques to act autonomously in dynamic and complex environments. The standard RoboCup league is using NAO humanoid robot [15] for competitions. NAO humanoid robot is developed by Aldebaran Robotics [14]. As shown in Fig. 1, it has 22 degrees of freedom. Cyberbotics also provides a simulation Tool for NAO robot called Webots [12].

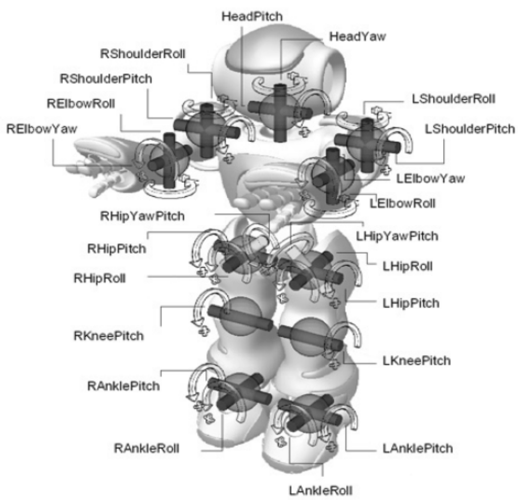

Fig. 1 – NAO Humanoid Robot with 21 Joints.

2.1. CASE-BASED REASONING

CBR is a reasoning methodology that simulates human reasoning by using past experiences to solve new problems, it is a recent approach to problem solving and learning.

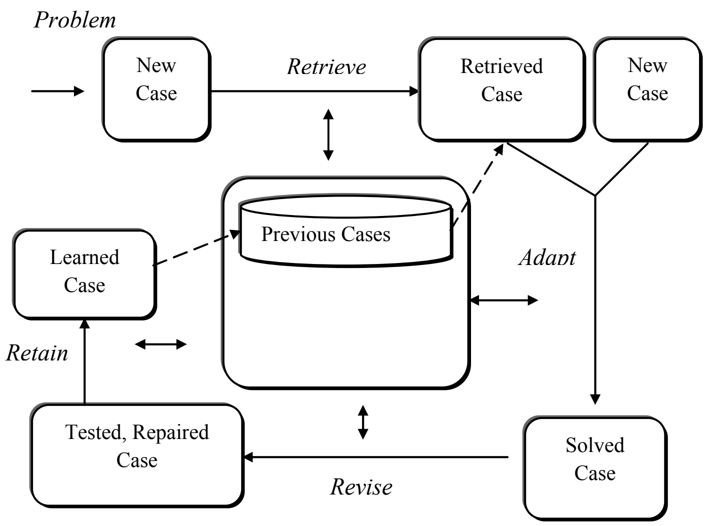

Fig. 2 – CBR cycle.

Agudo and Waston [5] shows how CBR models human reasoning, they describe CBR as a four-step process: retrieve the most similar case or cases; reuse (Adaptation) the information and knowledge in that case to solve the problem.

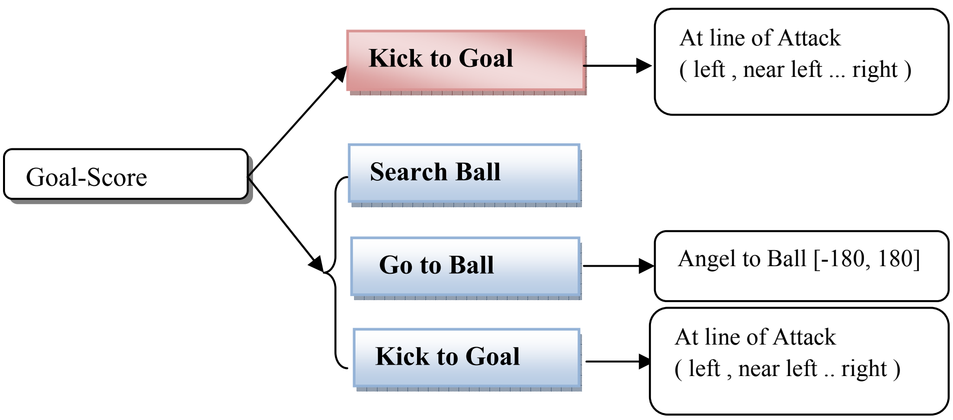

Fig. 3 – Goal-Score Complex Behavior consists of simple behaviors
and final robot localization at line of.

2.2. EXTENDED KALMAN FILTER

Kalman filter (KF) is widely used in studies of dynamic systems, analysis, estimation, prediction, processing and control. Kalman filter is an optimal solution for the discrete data linear filtering problem. KF is a set of mathematical equations which provide an efficient computational solution to sequential systems. The filter is very powerful in several aspects: It supports estimation of past, present, and future states (prediction), and it can do so even when the precise nature of the modelled system is unknown. The filter is derived by finding the estimator for a linear system, However, the real system is non-linear, Linearization using the approximation technique has been used to handle the non-linear system. This extension of the nonlinear system is called the Extended Kalman Filter (EKF) [1].

EKF have been extensively used in many applications where non-linear dynamics are prevalent. There are many instances where EKFs [1] have been used in different RoboCup leagues, *e.g.*, robot self-localization as well as for ball tracking [2–3]. In our research, we are focusing on the effectiveness of better ball tracking in the 3D simulation league to improve the extended subtasks such as Goal-scoring scenarios.

3. PROPOSED HYBRID PROBABILISTIC CBR-NEURAL ADAPTIVE ALGORITHM

Case adaptation plays the most crucial part in CBR systems. It means reusing previous experiences to execute new behaviors in the current situation. In this section, a new hybrid adaptation model has been developed for behavior control of Humanoid Soccer Robot. It is a modification to the previous HCBR [12], where adaptation rules are replaced by neural networks (NN's) learning and based on probabilistic ball tracking. It is a hybrid algorithm that combines adaptation and kalman ball tracking and NN's techniques. The coming sections describe the proposed algorithm in details.

The main algorithm of the proposed Probabilistic CBR-Neural adaptive behavior control is shown in Fig. 4. As shown, case adaptation is performed in a top-down fashion, where *Robot_Role* level is for adapting robot role, the *Robot_Skill* level is for adapting robot skill, the *Robot_Behaviors* level is for adapting robot behaviors and the *Robot_Reactive* level is for adapting the values of primitive behaviors.

STEP A: Apply EKF for Ball tracking and Goal-Scoring behavior learning cases.

STEP B: Execute the CBR-Neural Behavior Control for Case Adaptation

1. Input Real-time new *Role_case*.
2. Retrieve most similar *Role_case.*
3. Use ***Adaption Rules*** to Adapt the solution of the *Role_case*.
4. Append the solution of the new *Role_case* to the next level.
5. Input the real-time new *Skill_case*.
6. Retrieve most similar *Skill_case*.
7. Use ***Adaptation Rules*** to adapt the solution of the *Skill_case*.
8. Append the solution of the new *Skill_case* to the next level.
9. Input the real-time new *Behavior_case*.
10. Retrieve most similar *Behavior_case*.
11. Train and Adjust ***Neural Network*** on adaptation rules to adapt the solution of the retrieved *Behavior_case*.
12. Append the solution of the new *Behavior_case* to the next level.
13. Input the real-time new *Primitive_case*.
14. Retrieve most similar *Primitive_case*.
15. Train and Adjust ***Neural Network*** on adaptation rules to adapt the solution of the retrieved *Primitive_case*.
16. Execute the final adaptive behavior solution of the new *Primitive_case.*

Fig. 4 – The CBR-Neural Adaptation Algorithm for Robot Behavior Control Based on Kalman Filter Ball tracking and localization module of the RoboCup.

3.1. BALL TRACKING FOR BUILDING ROBOT CASES

We have implemented the following modelling criterion to capture the non-linear dynamics of the ball: first, we have conducted two rotations and a translation to move the perceived vision information to a fixed coordinate frame relative to the robot's torso; second, we have developed EKF models for **X** and **Y** axis separately. The ball state is given by:

$$\begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + \dot{x}_{t-1}\Delta t \\ \dot{x}_{t-1} \end{bmatrix} + \epsilon_t,$$

where, *t* is the index of the sampling interval. X is the position of the ball in the X-axis is the velocity of the ball along the X-axis. Λ t is the time step size. The measurement model is given by:

$$z_t = \begin{bmatrix} 1 & 0 \end{bmatrix} x_t + \delta_t .$$

We have used the following prediction model

$$\begin{bmatrix} x_t \\ y_t \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} cos(-\Delta\theta_t) & -sin(-\Delta\theta_t) & 0 & -\Delta x_t \\ sin(-\Delta\theta_t) & cos(-\Delta\theta_t) & 0 & -\Delta y_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ 0 \\ 0 \end{bmatrix},$$

where, **ΛX** and **ΛY** is the odometry translation [3, 4] and Λɵ is the odometry rotation. The Z-axis is ignored because we are interested in the 2D plane surface of the RoboCup. The update cycle of EKF is performed every 30 Frame. When the ball is not seen by the robot then the robot executes the search for ball behavior till it finds it. Many experiments are conducted for ball tracking. Figure 4 shows sample of performed experiments.

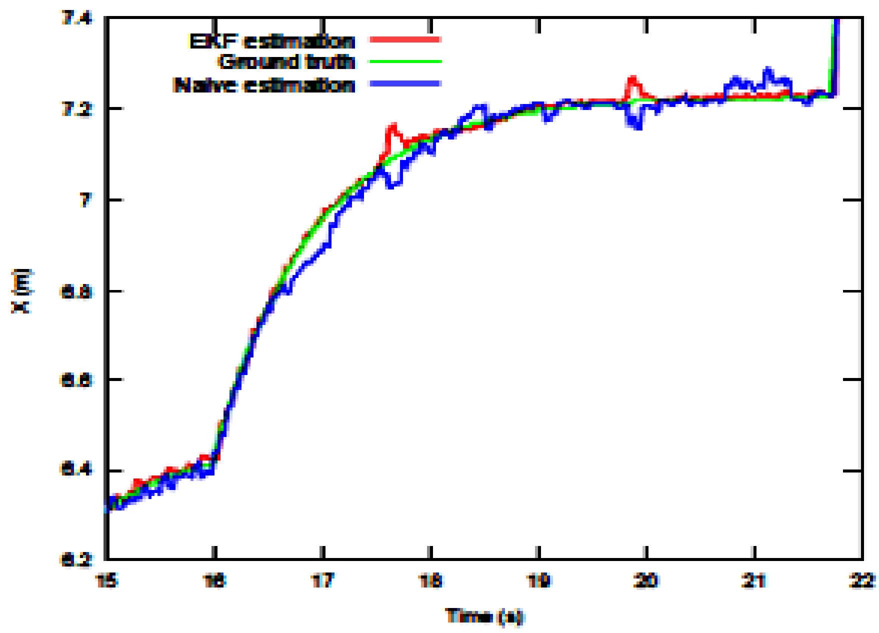

Fig. 4a – Experiment 1 – ball localization along *x*-axis.

The importance of ball localization and tracking described in section 3 is very important for the Robot to keep track for the ball and to learn the execution of important behaviors in real-time such as Goal-Scoring. The coming section describe step B of the proposed algorithm, where case adaptation is done by adaptation rules.

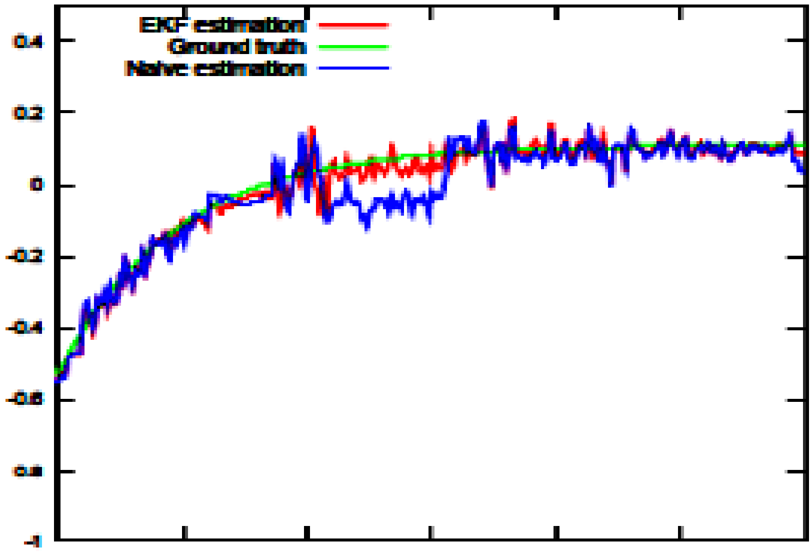

Fig. 4b – Experiment 1 – ball localization along *y*-axis.

3.2. CASE ADAPTATION BY ADAPTATION RULES

Case adaptation plays the most crucial part in CBR systems. It means reusing previous experiences to execute new behaviors in the current situation. Adaptation is usually done by Adaptation rules [11]. Adaptation rules means the rules that describe how the differences between the features of the new case and the features of the retrieved case affect the differences in their solution. As shown in Fig. 2, for example, in the RoboCup domain, the robot can use adaptation rule to change his kick left or kick right behavior of the ball according to the current situation.

An important informal example of one of our adaptation rules is:

"IF the feature ***Robot_x*** in the new case is ***1432*** and the similar feature ***Robot_x*** in the retrieved case is ***1200*** and the feature ***Robot_y*** in the new case is ***–840*** and the similar feature ***Robot_y*** in the retrieved case is ***20*** and the feature ***Goalie_x*** in the new case is ***-3130*** and the similar feature ***Goalie_x*** in the retrieved case is **-2090** and the feature ***Goalie_y*** in the new case is ***59*** and the similar feature Goalie_y in the retrieved case is 120 and the behavior solution in the retrieved case is kick_right. THEN the behavior solution of the new case is kick_far_left."

Our general form of the Adaptation Rule is:

IF *(N1 is value1, R1 is value1 ... Ni is value i, Ri is value j....... Nn is value n; Rn is value m)* ***AND*** *Retrieved_Behavior Solution*

THEN *New _Behavior Solution*

Where, *Ni* is the new feature i in the new case. *Ri* is the corresponding feature *i* in the retrieved case. *Retrieved_Behavior* is the behavior solution of the retrieved case. *New_Behavior* is the new behavior adapted for the new case.

However, still the limitations of applying hand-coded IF-Then rules or behavior state automata [7,8], which do not give the robot any experiences about current situation. In the coming section, we propose a neural network module to learn adaptation rules.

3.3. CASE ADAPTATION BY NEURAL NETWORKS

Our previous experiments using adaptation rules show low accuracy rate at the last two levels [21]. This is due a huge number of adaptation rules is needed to encode primitive behaviors. In this paper, we use our previous algorithm for case adaptation by using neural networks [10]. Our main goal is to use the learning capability of NN to learn adaptation rules and thus reduce the overall complexity. Our modified NN adaptation algorithm [10] is shown in Fig. 5. The main idea of our algorithm is to train the NN on IF-THEN adaptation rules and then use the trained NN to perform the case adaptation step. As shown in step 3 of Fig. 5, the adaptation rules at each level must be mapped into binary or numeric values in order to train the NN. Our general representation form of the mapped rules is:

$$[\,\mathbf{N1}, \mathbf{R1} \;\ldots \mathbf{Ni}, \mathbf{Ri} \ldots\ldots \mathbf{Nn};\ \mathbf{Rn},$$
$$\mathbf{Retrieved}_{\mathbf{behavior}},\quad \mathbf{Adapted}_{\mathbf{behavior}}\,]$$

where:

$\mathbf{N_i}$, is the mapped value of new feature *i* in the new case;

$\mathbf{R_i}$, is the mapped value of feature *i* in the retrieved case;

Retrieved_Behavior is the mapped behavior solution of the retrieved case;

Adapted _Behavior is the mapped behavior adapted for the new case.

Table 1

The MLP topology of the CBR-Neural Model

	MLP ***Robot_Behaviors*** **Level**	**MLP** ***Robot_Primitives*** **Level**
Input layer neurons	7	6
Hidden layer neurons	4	4
Output layer neurons	5	5
Learning rate	0.1	0.1
Momentum	0.7	0.7
Activation Function	Tansh	
The Adaptation rules no. trained on.	10	31
Root Mean Square Error.	0.0001	

The NN's are then trained on these mapped rules to learn how to make adaptation. We use two NN for the Robot_Behaviors level and the Robot_Primitives level. The NN type used at the Robot_Behaviors level is the feedforward multi-layer perceptron (MLP) [11] with one hidden layer that is trained with the backpropagation algorithm. Many experiments are necessary to choose the right rules to adjust the MLP of this level. After a number of trials, the topology of the MLP at which it has better performance is described in Table1.

Similarly, the NN type used at the Robot_Primitives level is MLP. The results of MLP's for Case adaptation are shown in Table 2.

4. EXPERIMENT RESULTS

In our cross-validation test [13], we use 820 cases stored in the case-memory and 80 cases are used for testing the performance accuracy. All the cases are tested for the behaviors of Attacker NAO. These cases are further classified into two groups, which are the *Goal-Score* and the *Dribble* cases. Table 2, shows the accuracy rate of our CBR-Neural Adaptive behavior control. As shown, we achieve a very high accuracy rate this is due to the following main factors:

Table 2

The accuracy rates of our CBR-neural for case adaptation

LEVEL	**No. of test cases classified by our controller**	**Adaptation Rules & NN's**	**Accuracy Rate**
Robot_Role	820	4	100%
Robot_Skill	812	12	93.3%
Robot_Behaviors	809	NN trained on 20 rules	90.83%
Robot_Primitives	810	NN trained on 34 rules	92.5%

1. Implementation of the EKF for ball tracking for the humanoid that enhances the overall learning process of the proposed CBR Behavior controller.
2. The usage of a fixed set of adaptation rules and testing them on the Goal-Score behavior only.
3. The use of neural networks to learn adaptation rules increases the accuracy rate.

5. CONCLUSION AND FUTURE WORK

This study illustrates a new adaptive behavior control for humanoid soccer robot based on probabilistic ball localization and tracking. It is based on case-based reasoning and neural networks techniques. The main aim of this research is to develop an adaptive behavior control for humanoid soccer robot. Thus enables the robot to be fully autonomous and adapt its 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 reduces the complexity of overall behavior execution in real-time. A high performance is achieved at all the levels due to the learning capability of neural networks. In future work, the controller will be tested using rest of soccer behaviors for goalie and defender players. Also, other similarity formulas will be tested like fuzzy similarity measures [20] to improve case retrieval.

REFERENCES

1. Sakai A., Kuroda Y., *Discriminatively Trained Unscented Kalman Filter for Mobile Robot Localization*, Journal of Advanced Research in Mechanical Engineering, *3*, 2010.
2. Laue T., Hass J.T, Burchardt A., Graf. C, Rofer T., Hartl A., and Rieskamp A. *Efficient and Reliable Sensor Models for Humanoid Soccer Robot Self-Localization*, Proceedings of the 4th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conf. on Humanoid Robots, Paris, 2009, pp. 22–29.
3. Huang R., Patwardhan, S. C., and Biegler L. T., *Robust extended kalman filter based nonlinear model predictive control formulation*, Proceedings of the 48th IEEE Conference on 28th Chinese Control (CDC/CCC) Conference, 2009.
4. J. Kim, Y.T. Kim, and S. Kim, *An accurate localization for mobile robot using extended kalman filter and sensor fusion*, in IJCNN, IEEE, 2008, pp. 2928–2933.
5. Agudo D.B., Waston L., *Case-Based Reasoning Research and Development*, Proceedings of 20th International Conference, ICCBR 2012, Lyon, France, September 3–6, 2012.
6. Celeberto, L.A., *Reinforcement Learning with Case-Based Heuristics for RoboCup Soccer Keepaway*, Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), Brazilian, 2012.
7. Berger R. and Lämmel G., *Exploiting past experience case-based decision support for soccer agents*, Advances in Artificial Intelligence, Volume 4667 of Lecture Notes in Computer Science, Springer, 2007, pp. 440–443.
8. Arcos Lluis, J. Mantaras, L. Ramon, Sierra Carles, Ros Raquel, *A CBR system for Autonomous Robot Navigation*, Proceedings of CCIA'05, Fronteries in Artificial Intelligent and Applications, pp. 299–306, 2005.
9. Urdiales, J. Vázquez-Salceda, E.J. Perez, Sànchez-Marrè, and Sandoval, F., *A CBR Based Pure Reactive Layer for Autonomous Robot Navigation*, 7th IASTED International Conference on Artificial Intelligence and Soft Computing, Banff, Canada, 2003, pp. 99–104.
10. Espinosa, R.R., Veloso, M., *Executing Multi-Robot Cases Through a Single Coordinator*, 6th AAMAS'07 International Joint Conference on Autonomous Agents and Multi-Agent Systems, Honolulu, Hawaii, 2007.

11. El-Bagoury, M. B., Salem M. A.B., Burkhard, D., H., *Hierarchical Case-Based Reasoning Behavior Control*, Annals of University of Craiova, Math. Comp. Sci. Ser. **36**, *2*, 2009.
12. Burkhard, H.-D., Duhaut, D., Fujita, M., Lima, P., Murphy, R., Rojas, R.: *The road to RoboCup 2050*, IEEE Magazine of Robotics and Automation, **9**, *2*, pp. 31–38, 2002.
13. Abdel-Badeeh M. Salem, Bassant M. El Bagour. *A Case-Based Adaptation Model for Thyroid Cancer Diagnosis Using Neural Networks*, Proc. of the 16th International FLAIRS conference, Florida, U.S., 2003, pp. 155–159.
14. * * *, Aldebaran Robotics http://www.robotshop.com/aldebaran-robotics-nao-h25-humanoid-robot-academic-3.html
15. * * *, NAO-Humanoid-Robot http://www.robocup.org/2012/12/call-for-participation-robocup-german-open-2013/