Impossibles AIBO Four-Legged Team Description Paper RoboCup 2006

Supervisor: Jafar Habibi
Hamid Reza Vaezi Joze, Saman Aliari Zonouz
Siavash Rahbar, Majid Valipour, Alireza Fathi
Department of Computer Engineering
Sharif University of Technology
{habibi, aliari, vaezi, rahbar, valipour, a_fathi}@ce.sharif.edu

Abstract

AIBO four-legged soccer league is considered as one of the most popular test-beds for Artificial Intelligence (AI) and Robotics. This paper presents Impossibles AIBO 4-Legged main architecture for RoboCup 2006 which is going to be held in Bremen, Germany. This architecture includes different modules such as World Modeling (WM) module, Vision (SVS and GVS), Action (Decision Making and Motion Controller), Communication, and Localization modules. These modules are explained briefly in this paper. Some parts of our architecture are being implemented; therefore, we have employed the corresponding modules of other teams. For instance, we are using UPENALIZERS's trajectories. As explained below, our main objective is to be ranked as one of the top four teams in RoboCup_2006. "Impossbiles" is the first Iranian team participating in AIBO 4-legged competitions. There were no such local competitions in Iran; hence, we have not been able to participate in such contests; however, we believe that "Impossibles" strong background in RoboCup, the uploaded film, and Team Report prove our endeavor in Artificial Intelligence and Robotics Laboratory (AIRL).

1. Introduction

As a research group, *"Impossibles"* team has been set up in Artificial Intelligence and Robotics Laboratory (AIRL) of Computer Science and Engineering Department at Sharif University of Technology since March 2004. Research Areas of *"Impossibles"* were categorized into three groups including Artificial Intelligence (Machine Learning, Multi-Agent Systems, and Reasoning), Theoretical Computer Science (Algorithms, and Data Structures), Soft Computing (Fuzzy Theory, and Genetic Algorithms).

Having done the background researches, all of the members decided to exploit their knowledge in a practical and real world environment. Since several teams from Sharif University of Technology had achieved noticeable successes from RoboCup2000 in Melbourne to RoboCup2003 in Padua, RoboCup was selected as the first choice; therefore, we were able to employ their corresponding experiences. Table 1 demonstrates a brief overview of these achievements.

As explained above, RoboCup's interesting features attracted us to begin implementation of our previously designed ideas in Rescue Simulation Environment (RSE) to participate in RoboCup2005 in Osaka. So it was our first participation in such international competitions. Having coded from scratch, we applied our new ideas. Consequently, *"Impossibles"* got world championship in Rescue Simulation League in Osaka 2005.

Once world championship was achieved, team members made decision on continuing their research objectives through AIBO 4-legged League. AIBO League was preferred over the other RoboCup Leagues because of the following four reasons which are also considered as *"Impossibles"* objectives in AIBO league. AIBO does support the real world challenges, whereas Rescue Simulation does not. Additionally, it is the only physical robot league in which there is no need to get involved into mechanical aspects of the robots' design, so it was the most similar league to the simulation leagues such as Rescue Simulation. Furthermore, AIBO 4-legged league supports most of the research interests of the team members such as machine learning. Lastly, several highly ranked universities (e.g. CMU and Texas-at-Austin) have done research on various branches of AI using AIBO robots; hence, it is thought to be a qualified infrastructure for our team members to do research on. On the other hand, we follow our competitive objective which is to be ranked as one of the first four teams of AIBO league in Bremen, Germany, 2006.

Since Vision and Image Processing were required in order to accomplish the AIBO project, defined in Artificial Intelligence and Robotics Laboratory (AIRL) of Computer Science and Engineering Department at Sharif University of Technology, we came to conclusion to invite some of the members of Vision Group at IPM School of Mathematics Scientific Computing Center.

Table 1: History of Sharif Teams in RoboCup

Year	Team	League	Rank
RoboCup 2000 Melbourne	Sharif CE	Middle Size	Third
RoboCup 2001 Seatle	Arian	Rescue Simulation	Second
	Sharif CE	Engineering Challenge	First
RoboCup 2002 Fukuoka & Busan	Arian	Rescue Simulation	First
RoboCup 2003 Padua	Arian	Rescue Simulation	First
	CEDRA	Rescue Robot	Second
RoboCup 2005 Osaka	**Impossibles**	Rescue Simulation	First
RoboCup 2006 Bremen	**Impossibles**	AIBO 4-Legged	???

2. Architecture

Our previous experience in Multi-Agent System (MAS) architecture design in Rescue Simulation Environment leaded us to World Model Based Architecture (WMBA). Having made some subtle modifications in WMBA, we employ it as our basic design architecture for concurrently-running objects of Open-R SDK. WMBA contains three major tasks that are being done independently in three subsystems: Sensors, Communication, and Action subsystems. These subsystems are performing in a way that objectives are achieved and constraints are convinced. The main constraint of the AIBO robots are the limited resources such as CPU and 500Kbps limit on data transmission in wireless communication. Figure 1 demonstrates the World Model Based Architecture (WMBA).

Sensing subsystem is responsible for perception via camera and other sensors. Additionally, communication subsystem is employed to transmit information among AIBO robots. Furthermore, action subsystem is in charge of determining what the AIBO robots decide and perform. Decision Making (DM) is responsible for high level decision makings, whereas in Motion Controller (MC) low level skills are implemented. Last of all, Localization is considered as an input gate to World Model (WM). Localization's main task is updating World Model (WM) using the data received from the adjacent subsystems, i.e. Motion Controller (MC), World Model (WM), Communication, and Vision.

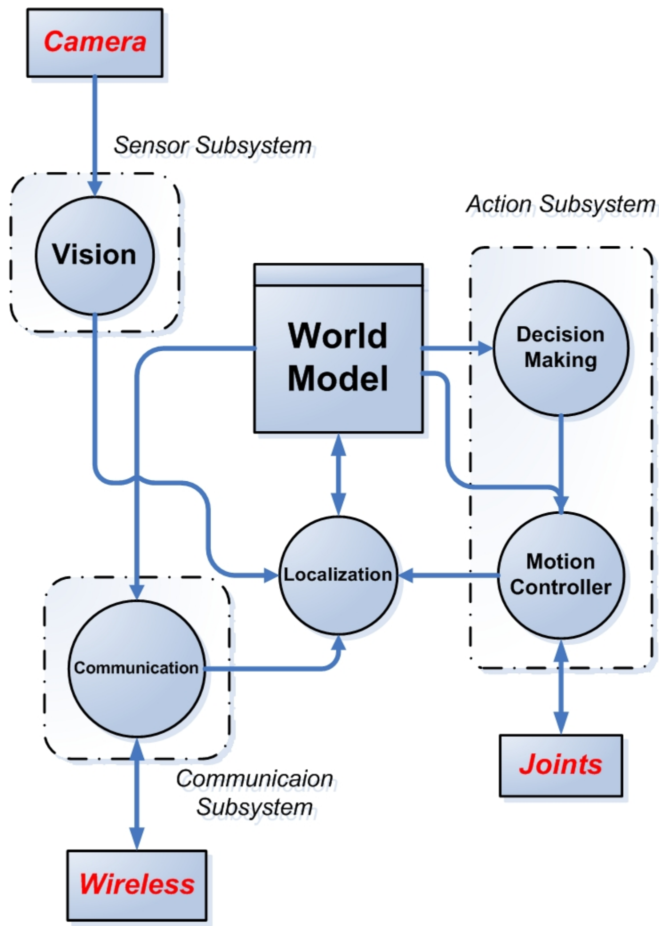

Figure 1: Impossibles AIBO Architecture

3. Fuzzy World Modeling

In a real world robotics environment such as AIBO 4-legged league, agents have to have interactions with several physical objects, e.g. the orange ball. This interaction is typically implemented as a perception-action loop. AIBO Robots are equipped with sensors that perceive physical characteristics of the environment and they use these percepts to build an internal representation of the environment, i.e. World Model (WM). Once this world model is built, agents exploit it in order to accomplish their assigned tasks. Generally, the WM anchoring process consists of the following three steps:

Classification: each perceived object (i.e. set of features produced by a sensor) is classified according to the predefined features of known objects.

Fusion: Objects perceived by different sources, i.e. agents, that can be associated to the same physical object are merged. Evidence theory is employed.

Tracking: The perceived information via current inputs update the corresponding objects' features in the world model. We assume that smart sensors produce sets of features, where each feature is a triple: $\langle label, v, \rho \rangle$. The $label$ of a feature is its name, v is its numerical value, and ρ is its reliability value, i.e. how the data is assumed to be reliable given the specific sensor and the acquisition situation.

(1) If the perceived instances do not match any instance in the world model, a new instance is created with the value of the perceived instance. (2) If an instance in the world model does not match any perceived instance, the reliability values of its attributes are exponentially decreased by a coefficient between zero and one. (3) If a perceived instance matches an instance in the world model, their reliability values are composed by the arithmetic mean.

4. Communication

a. Centralized vs. Distributed

Generally, we consider the communications amongst players distributed, but due to the large amount of transmitted data and hence time-consuming processes, agents themselves accomplish their own jobs and broadcast the results, i.e. processes data.

If there wasn't any broadcast feature in our access media, having centralized communication might also reduce number of messages which are needed to share all information among agents.

m = number of messages needed to have all information shared between agents

- With broadcast message:
  - Centralized approach – $m = n + 1$
    [n peer to peer message + 1 broadcast]
  - Distributed approach – $m = n$
    [n broadcast message]
- Without broadcast message:
  - Centralized approach – $m = n + n = 2n$
  - Distributed approach – $m = n \times (n - 1)$

When we are considering our access media properties including its broadcast ability and limited bandwidth and also the fact that defining an agent as center might be unreliable we decide to use distributed communication by broadcasting messages. The messages contain low level data sensed and acquired by agents from the surroundings such as ball, teammates, and opponent players which are used in localization and updating word model in with each agents self awareness.

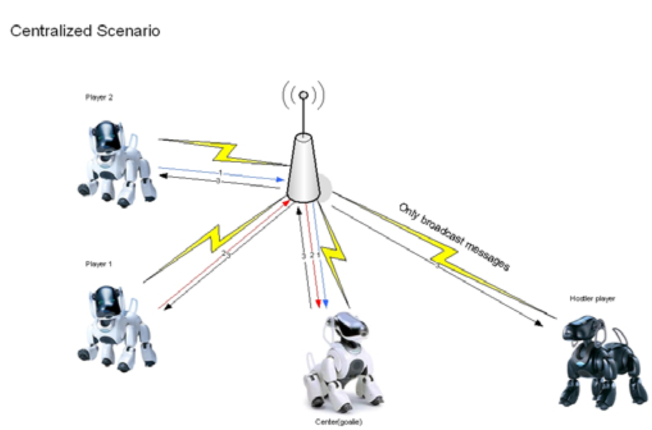

Figure 2: Centralized Scenario

b. UDP vs. TCP

Selecting either UDP or TCP is thought to be the primary task, and according to **'NS2'** simulation result for both UDP and TCP scenarios and other teams hints we decide to use UDP, because of lesser overhead in comparison with TCP and ability of broadcasting by UDP which is essential for us to minimize our number of sent messages in our distributed strategy.

NS2 simulator have also been employed to simulate UDP data transfer in wireless mobile networks, in order to select optimized value for our UDP packet size to achieve maximum bandwidth considering possible data collision and opponent team inference. The following figure demonstrates a snapshot of our simulated situation.

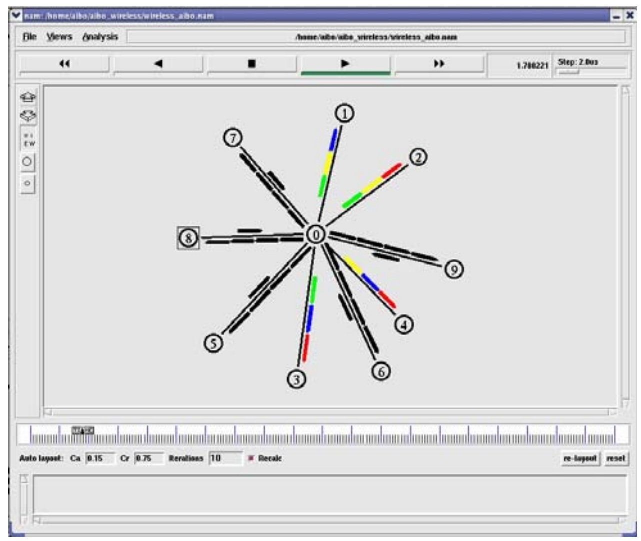

Figure 3: NAM Snapshot of Network Simulator

In our simulated situation, there is just one access point [node #0], four players [nodes #1 2 3 4], and five other network traffic producers (4 hustler players and one manager). Also the simulated wireless network implements multicast packet switching and 802/11 MAC protocol and random movement for players.

5. Localization

Unlike other processes running without being blocked, Localization is not in active state until it receives a message from Vision, Motion Controller (MC), or Communication. The most important information is the position of robot itself which by Self-Localization module. Other data such as locations of other robots and ball are determined using Object-Localization.

a. Self-Localization

Self-Localization module takes the previous positions and differential locations as its input. Having processed the inputs, it then updates the World Model. Generally, positions are stored as (x, y, θ) triples that are 2D position of robot and its direction.

Although, the most popular approach for position estimation of mobile robots is Monte Carlo Localization (MCL) that was widely being used by 4-Legged AIBO soccer teams, we need a method that is compatible with our fuzzy probabilistic world model and also is able to support real time applications. We present a new approach that is a probabilistic approach for mobile robot localization.

i. Probabilistic Distribution Localization (PDL):

It considers a PDF for each variable (such as x, y and θ for AIBO). In Monte-Carlo Localization (MCL), samples are stored by (x, y, θ) triples and a weight factor (p>0). In contrast, in Probabilistic Distribution Localization (PDL), we have three PDF for each sample (one PDF for each of x, y and θ). Also each differential motion, i.e. (Δx, Δy, Δθ), contains three corresponding PDFs. So we need to update the PDFs after movement update (from Motion Controller) and sensor update (from Vision).

Movement Update: We consider 'X' a random variable for probabilistic distribution of 'x' position and 'ΔX' as a random variable for probabilistic distribution of movement of 'x' so the new value for 'X' will be 'X+ΔX'. In this way the corresponding PDF for x is obtained.

Sensor Update: As mentioned above, each perceived data by vision module in *"Impossibles"* software contains a PDF for each variable for example 'x' and 'p' that is the probability that this sample is correct. Now we create a new PDF for 'x' by the following formula:

$$f_{Xnew}(x) = p \times f_{Xvision}(x) + (1-p) \times f_{Xold}(x) \qquad (1)$$

PDFs may become useless after too many movements or sensor updates with small 'p'. So "Sensor Resetting Localization" is employed which considers a threshold for average of 'p'. Some new samples must replace when it becomes lesser than the assigned threshold.

b. Object Localization

Object localization is responsible for collecting data and their reliability about objects' position from vision and communication to estimate these positions. We have employed *evidence theory* in order to estimate the locations of objects of interest on the field. Evidence theory begins with the familiar idea of using a number between zero and one to indicate the degree of support a body of evidence provides for a proposition, i.e. the degree of belief one should accord the proposition on the basis of the evidence. Evidence theory focuses on the combination of degrees of belief or support based on one body of evidence with those based on an entirely distinct body of evidence. The heart of the theory is Dempster's rule for effecting this combination.

6. Decision Making

Decision Making (DM) module in *"Impossibles"* AIBO robots have a layered architecture. In fact, DM module consists of two major layers: Team Behavior, i.e. tactics, and Individual Behavior i.e. the techniques employed by individual players. DM module gets its input from the system's world model accompanying with some degree of belief which is due to existing uncertainty in real system environments. Then the robot analyzes the input in a two-step procedure. Team behavior resolves the whole team behavior, e.g. tactics stored in a database (e.g. Figure 6). Then Individual behavior module obtains the whole team behavior and world model decides one of the possible actions for individual robots to do. As a matter of fact, these actions are the outputs DM. *"Impossibles"* actions (skills) are (1)shooting in a specified direction with a particular power, (2)blocking the way in a special direction, (3)walking through a path determined by an array of points, (4)looking in one direction, and (5)grabbing the ball.

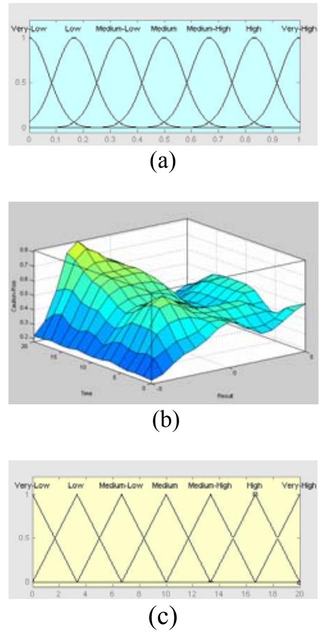

Figure 4: Fuzzy Controller
(a)fuzzification
(b) Rule Base
(c) Defuzzification

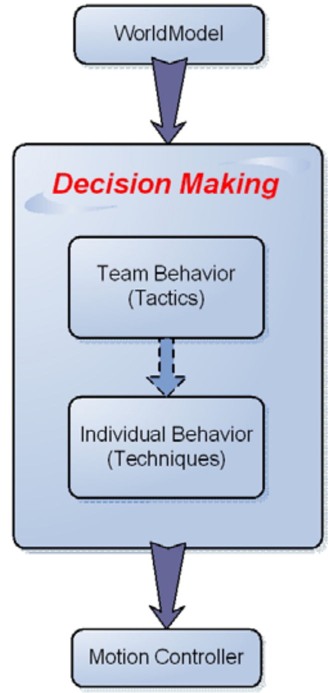

Figure 5: DM Architecture

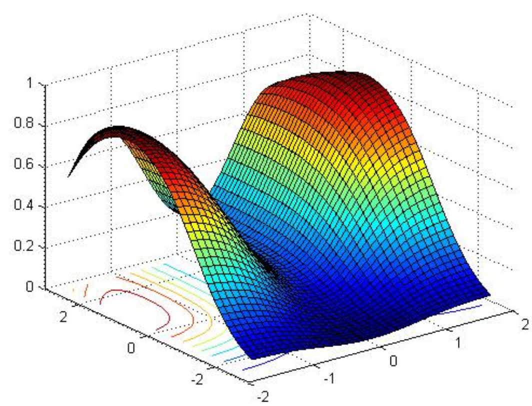

Figure 6: A sample Team Strategy

7. Vision

According to *"Impossibles"* AIBO architecture, each robot updates its world model using three inputs from Sensors, wireless communication, and vision modules. Vision in AIBO robots is in charge of receiving two inputs and producing a set of two outputs. These inputs and outputs are as follows:

Inputs: (1) A stream of images taken by robot's camera. Surely, these images contain a large amount of noise which has been caused by some issues such as robot's motion or distance of the objects in image from the robot's location in the field. (2) AIBO robots' sensors provide us with a set of joints' angles over time. So, direction of the camera and current condition of the robot is identified using this type of input.

Outputs: (1) Distances and angles to a fixed set of color-coded objects with known locations, which can be used to *localize* the robot on the field. (2) Distances and angles for a varying set of mobile objects.

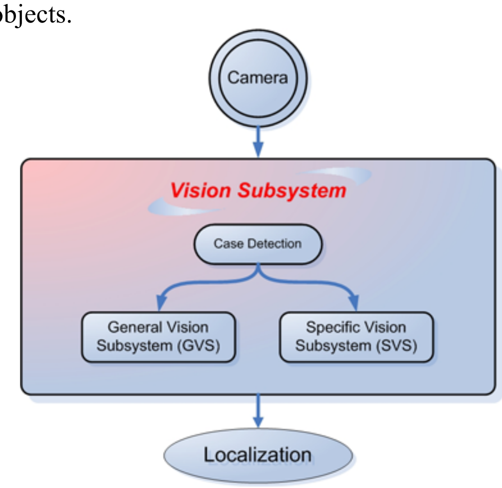

Figure 7: Vision Intra-Architecture

Generally, we employ two approaches: General Vision Subsystem (GVS), and Specific Vision Subsystem (SVS). The GVS algorithms are mainly based on the UT Austin Villa vision system. The following figure demonstrates an image in GVS process. SVS approaches cannot be employed generally by robots, because of their overtime-consumption. Hence, SVS approaches are used in special cases such as the case that vision subsystem receives a signal from self-localization module that it is unable to self-localize the robot.

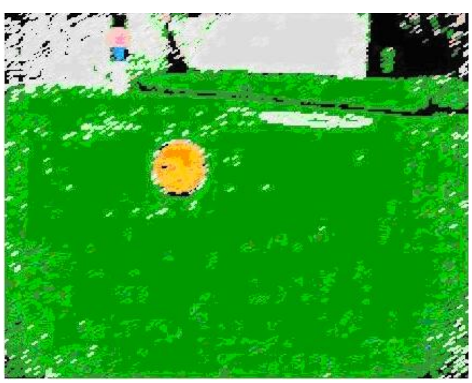

Figure 8: Region Segmented Image

Vision Module Architecture consists of three major subsystems. Case Detection (CD), General Vision Subsystem (GVS), and Specific Vision Subsystem (SVS) are the mentioned subsystems of VMA. First of all, the state of the AIBO robot is to be determined. State can be assigned one of the following values: Ordinary, Blocked, and post-Kidnapped. AIBO robots are usually in Ordinary state. In other words, robots are playing freely most of the time without other players interfering. GVS approaches are used in such situations in which robots have freedom of action. As a matter of fact, GVS approaches are employed in such cases because they are computationally cheap.

Besides, Blocked state is encountered in situation that the AIBO has failed to move after trying for some time. SVS approaches are run in these cases to realize the reason of being blocked. SVS is exploited, because GVS has failed to detect objects exactly in order to let DM module decide what to do properly. Additionally, post-Kidnapped state is happened in few moments. As a case in point, having booked, the robot is placed out of play for thirty seconds. In this status, the robot state is thought to be kidnapped. After repositioning on the field, the robot will make use of SVS approaches in the first moments to let the localization module self-localize exactly. Exact self-localization in the first few seconds of being repositioned on the field is an important factor. If the first self-localization is not done properly, the fault can be propagated until being in the situation that a land mark is recognizable by GVS.

8. Motion

"Impossibles" AIBO robots employ a Layered Motion Controlling approach. Motion Controller (MC) system consists of two submodules: Skills and Inverse Kinematics module. There are five skills available: Shooting, Walking, Looking, Blocking and Grabbing. These skills are Decision Making (DM) module output and are run concurrently as OPEN-R objects. Using concurrent Skills as separate modules makes development easier but has some disadvantages. As a case in point, adding a new skill requires some changes in World Model and Inverse Kinematics module.

Like Skills, Inverse Kinematics module is also an OPEN-R object. It sends commands to the joints and receives joint values. Self-Localization system also uses joint values; therefore, having received the joints' values, the Inverse Kinematics module sends the values of the joints to the Self-Localization subsystem. Skills Conflict Prevention (SCP) is also done by Inverse Kinematics submodule. As a matter of fact, it guaranties not to have conflicts amongst skills. When two skills are trying to simultaneously use a joint, Inverse Kinematics submodule selects the more important skill and reports failure to the skill with the lower priority.

a. Skills

Each skill has its own input parameters and uses specific information from World Model (WM). They are responsible for executing the received commands from Decision Making (DM) subsystem and reporting the state of the robot's joints when executing the command and notifying the Decision Making (DM) module when execution is finished. Of existing skills, shooting is discussed here.

Shooting skill is considered as one of the most important skills in the AIBO soccer environment. Different shooting methods differ in delay, speed, stability of the robot and accuracy. We classify our shooting methods into two groups: Controlled and Non-controlled shoots. In Controlled shoots robot gets the ball ownership completely before shooting it. In this type, motions of robot after grabbing ball are predefined and joint trajectories could be looked up from a Look Up Table (LUT). Although these shoots have more delay, they are accurate. The famous example of this type is the Chest shoot witch is widely used in competitions. Another example is UMIGAME that is a backward shooting method. Non-controlled shoots are faster but they are inaccurate and need good prediction of ball movement. They are employed in cases that the robot is far from ball and can't reach ball, e.g. because of obstacles. One of the most popular examples for this type is German Team's One-hand shooting method.

Shooting skills receives two parameters from Decision Making (DM) subsystem; direction and power. Currently the power parameter is ignored and robots always use maximum power for shooting. Chest Shoots are accomplished through a three-step process:

Hold the ball under chain: As beginning state of chest shooting.
Turn according to direction input: Then the robot should turn to the desired direction without releasing the ball. This is just like normal turning. The most important difference is that robot should hold the ball under its chin.
Kick the ball: The last part is the final shooting step. Joint trajectories for this part is predefined and are tuned for maximum speed of the ball after pushing.

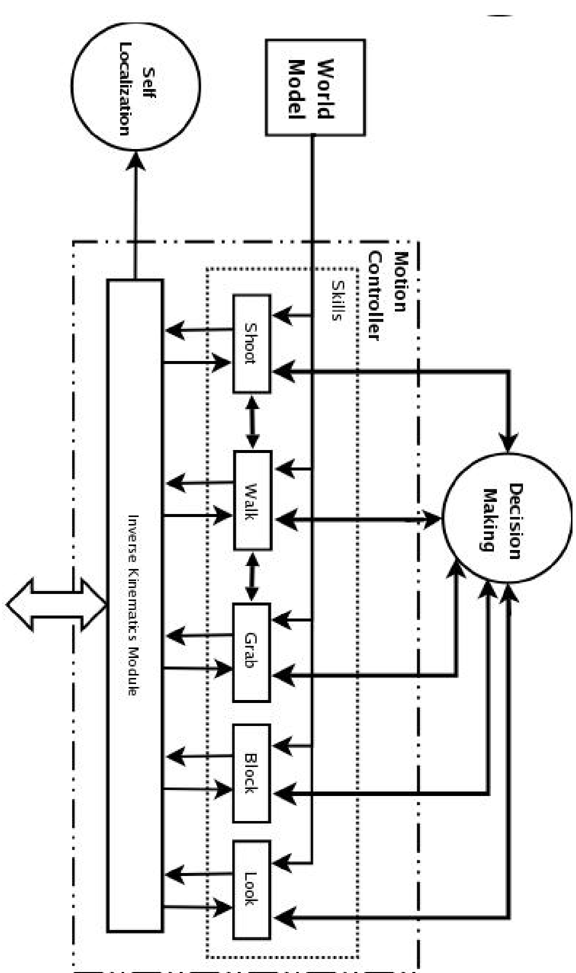

Figure 9: Motion Intra-Architecture

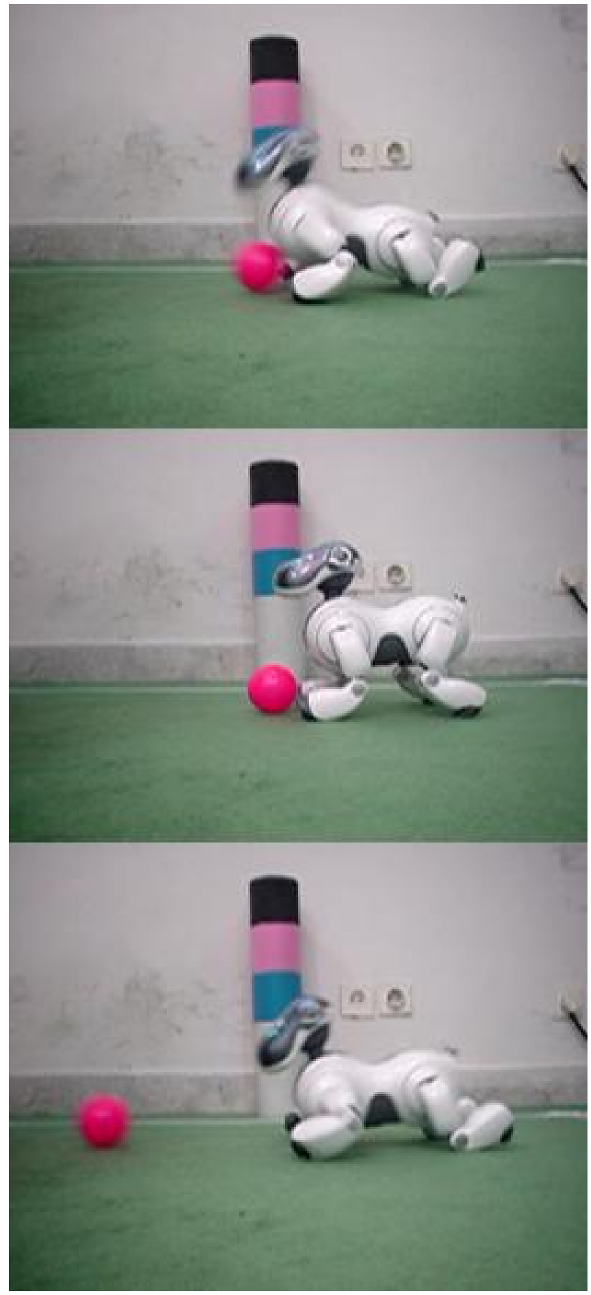

Figure 10: Chest Shooting

9. Acknowledgements

We would like to appreciate P. Stone, C. Lucas, and S. Bagheri for their valuable comments on Reinforcement Learning, Emotional Control, and Fuzzy Controllers respectively.

10. Conclusions

In this paper, the *"Impossbiles"* AIBO architecture and its subsystems were explained briefly. Additionally, we presented *"Impossibles"* research group's backgrounds in RoboCup (e.g. World championship in Rescue Simulation League), and our competitive and research objectives. As the first Iranian team participating in RoboCup_2006 AIBO 4-legged League, we intend to be ranked as one of the top four teams. As research point of view, we are implementing BELBIC (Brain Emotional Learning Based Intelligent Controller) by C. Lucas on AIBO robots.

References

[1] http://openr.aibo.com.

[2] http://www.tzi.de/4legged.

[3] Jafar Habibi, Saman Aliari Zonouz, Hamid Reza Vaezi Joze, Majid Valipour, Moahammad Reza Ghodsi, Alireza Fathi, *"Impossibles Team Description"*, RoboCup, Osaka, Japan, 2005.

[4] Thomas Rofer et al., "German Team Technical Report for RoboCup 2005", RoboCup, Osaka, Japan, 2005.

[5] Peter Stone, Peter Stone, Kurt Dresner, Peggy Fidelman, Nate Kohl, Gregory Kuhlmann, Mohan Sridharan, and Daniel Stronger, "The UT Austin Villa 2005 RoboCup Four-Legged Team", RoboCup, Osaka, Japan, 2005.

[6] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localization: Efficient position estimation for mobile robots", In Proceedings of the National Conference on Artificial Intelligence, 1999.

[7] S. Lenser, J. Bruce, and M. Veloso, "CMPack: A complete software system for autonomous legged soccer robots", In Autonomous Agents, 2001.

[8] Glann Shafer, A mathematical theory of evidence, Princeton University Press, 1976.

[9] Johann Heinrich Lambert, "Neues Organon", 1764.

[10] Gray, A., "Modern Differential Geometry of Curves and Surfaces with Mathematica", Boca Raton, CRC Press, 1997.

[11] Hilbert, D. and Cohn-Vossen, S., "Geometry and the Imagination", New York: Chelsea, pp. 3-4, 1999.

[12] Saman Harati Zadeh, Ramin Halavati, Saeed Bagheri Shouraki, Caro Locus, "Emerging Simple Emotional States in Zamin Artificial World",

[13] Caro Locus, Danial Shahmirzadi, and Nima Sheikholeslami, Introducing a Controller Based on Brain Emotional Learning Algorithm: BELBIC (Brain Emotional Learning Based Intelligent Controller), International Journal of Intelligent Automation and Soft Computing (AutoSoft), August 2002, USA.

[14] J. Habibi, and P. Nayeri, "Centralized vs. Non-Centralized Decision-Making in Multi-Agent Environments", 11th CSI computer Conference, January, 2006.

[15] Peggy Fidelman and Peter Stone, "Layered Learning on a Physical Robot", February 2005.

[16] Nate Kohl and Peter Stone, "Machine Learning for Fast Quadrupedal Locomotion" In Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI), San Jose, CA, July 2004

[17] Mohan Sridharan, Peter Stone, "Real-Time Vision on a Mobile Robot Platform", IROS, 2005.

[18] Peter Stone, Kurt Dresner, Peggy Fidelman, Nate Kohl, Gregory Kuhlman, Mohan Sridharan, Daniel Stronger, "UT Austin Villa 2005 RoboCup Four-Legged Team Report", 4-Legged Soccer League RoboCup, Osaka, 2005.

[19] Brodley, C., and Utgoff P., "Multivariate decision trees. Machine Learning", 1995.

[20] Brown, T., and Koplowitz, J. 1979. The weighted nearest neighbor rule for class dependent sample sizes. IEEE Transactions on Information Theory 617–619.

[21] Jain, R., Kasturi, R., and Schunck, B., "Machine Vision". McGrawHill, 1995.

[22] Laboratories, N. 1999. Cognachrome image capture device. http://www.newtonlabs.com.

[23] Silk, E. 1999. Human detection and recognition for an hors d'oeuvres serving robot. http://www.cs.swarthmore.edu/ silk/robot/.

[24] FORSYTH D. A., PONCE J.: *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[25] Nearest Neighbor Pattern Classification, T. M. Cover, and P. E. Hart.

[26] Jeff Hyams, Mark W. Powell, and Robin R. Murphy. Cooperative navigation of micro-rovers using color segmentation. *In Journal of Autonomous Robots*, 9(1):7–16, 2000.

[27] B. W. Minten, R. R. Murphy, J. Hyams, and M. Micire. Low-ordercomplexity vision-based docking. *IEEE Transactions on Robotics and Automation*, 17(6):922–930, 2001.

[28] Ana Fred, and Anil K. Jain, "Evidence Accumulation Clustering based on the K-Means Algorithm".

[29] R. Bergevin and M. D. Levine. Generic object recognition: Building and matching coarse 3d descriptions from line drawings. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15:19--36, January 1993.

[30] York, D. "Least-Square Fitting of a Straight Line." Canad. J. Phys. 44, 1079-1086, 1966.

[31] Nello Zuech, "Understanding and Applying Machine Vision", Second Edition, Marcel Dekker, Inc.

[32] Becher, W. D., "Cytocomputer, A General Purpose Image Processor," ASEE 1982, North Central Section Conference, April 1982.

[33] Lougheed, R. M. and McCubbrey, D. L., "The Cytocomputer: A Practical Pipelined Image Processor," Proceedings of the 7th Annual International Symposium on Computer Architecture, 1980.

[34] Sternberg, S. R., "Language and Architecture for Parallel Image Processing," Proceedings of the Conference on Pattern Recognition in Practice, Amsterdam, The Netherlands, May 21–30, North-Holland Publishing Company, 1980.

[35] Sternberg, S. R., "Architectures for Neighborhood Processing," IEEE Pattern Recognition and Image Processing Conference, August 3–5, 1981.

[36] S. B. Niku, Introduction to Robotics; Systems and Applications, Jul 2001, Prentice Hall.

[37] OPEN-R SDK. http://www.openr.org.

[38] H. Fukumoto et al., Baby Tigers Dash: Osaka 4-Legged Robot Team, RoboCup, Osaka Japan, 2005.

[39] http://www.germanteam.org/tiki-index.php.

[40] M. S. Kim, and William Uther, "Automatic Gait Optimisation for Quadruped Robots", Proceedings of the 2003 Australian Conference on Robotics & Automation.

[41] J. Bruce, S. Lenser, and M. Veloso, "Fast Parametric Transitions for Smooth Quadrupedal Motion", IEEE Transactions on Robotics, April 2005.

[42] Nate Kohl and Peter Stone, "Policy Gradient Reinforcement Learning for Fast Quadrapedal Locomotion", In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, May 2004.

[43] S. Chernova, and M. Veloso, "An Evolutionary Approach to Gait Learning for Four-Legged Robots", In Proceedings of IROS'04, September 2004.