Control System Architecture for an Autonomous Quadruped Robot

Nikola Georgiev Shakev[1], Jordan Kirilov Tombakov[1], Ivan Kalvatchev[1], H. Levent Akın[2], Andon Venelinov Topalov[1], Petya Emilova Pavlova[1]
[1]TU Sofia, branch Plovdiv, Bulgaria
[2]Boğaziçi University Istanbul, Turkey

Keywords
Autonomous robot, Control system architecture, Locomotion, Object's color identification

INTRODUCTION

The Sony Legged Robot League is an international robot soccer competition that has been launched within the RoboCup initiative [1]. Sony's quadruped robot AIBO has been adopted as a hardware platform, so the competition in this league is on a software level. The competition rules do not allow remotely controlling robots in any way and the robots are entirely autonomous. Their onboard 64-bit RISK processor provides enough computational power to perform image processing, localization and control tasks in real time. The only information available for decision-making comes from the robot's onboard camera, built in proximity sensors and sensors reporting the state of the robot's body (like the built in acceleration sensor and gyroscope).
In this work, the architecture of a control system developed for the AIBO robotic dog is presented. It is a joint team effort of students and their professors from Boğaziçi University, Istanbul, Turkey and TU Sofia Plovdiv branch, Plovdiv, Bulgaria. The system is designed and applied to the Cerberus robotic team participating in the Sony Legged League within the RoboCup competition.

ARCHITECTURE

The main goal of the joint Turkish-Bulgarian team is to build a research platform, which allows robust and efficient carriage of quadruped AIBO robots playing soccer. A modular architecture has been adopted and implemented. The following are the main system components (modules): Vision, Planner, Behaviors, and Locomotion. The relationship between the modules is shown in Figure 1.

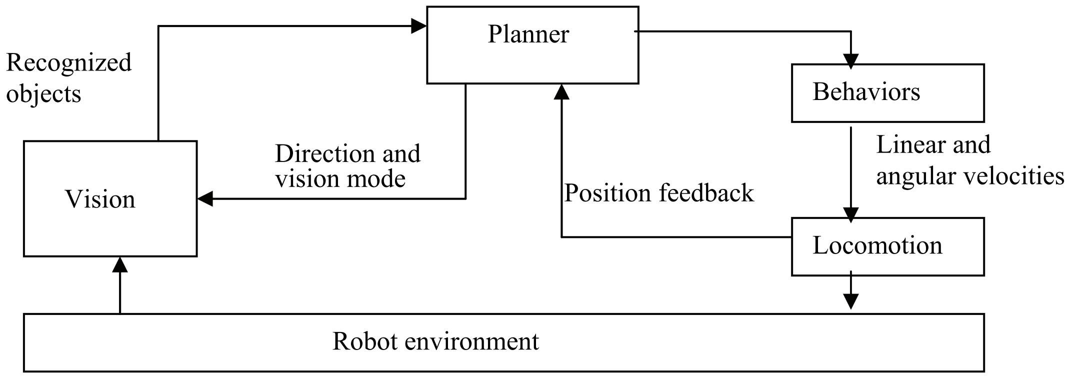


Fig.1

Vision and localization

The developed vision and localization module recognizes different objects on the playground (ball, teammates, opponent players, landmarks, goals etc.) and helps to find their and AIBO's own position on the field. The image processing algorithms designed the robotic applications have to comply with the requirement for a real time operation. This is usually achieved by processing binary images that permit fast segmentation of the viewing field. In the case of color images captured by AIBO's vision system binary methods can be applied after a preliminary color separation of the pixels. The latter could be obtained using three different types of analysis – (i) in the space of the *RGB* reproducing signals, (ii) directly in the space of the input complex TV signal or (iii) in the color space described by a triple consisting of hue – *H*, saturation – *S* and achromatic parameter. The first two approaches permit hardware implementation with signal processors but the problem is that their implementation frequently causes an overlap (overlay) effect between the ranges of object's definitions as a consequence. The latter leads to problems in automatic object separation and involves a complicated description of the color using combination of clusters [2]. By contrast, the third approach can deal easily with uncovered borders of the ranges of different colors. The search for objects by using *H* and *S* parameters is similar to the human perception and understanding of colors. However, the method involves during its implementation an additional transformation which makes it computationally intensive [3, 4].

It was one of the aims of this work to develop fast algorithm performing real time color segmentation on the basis of chromaticity of the pixels composing objects' images. In the developed algorithm each frame is transformed from the three dimensional – *YUV* matrix derived from the robot's built in TV camera to one-dimensional one composed by marks of different color blobs. Figure 2 shows the implemented data processing. Fast *HS* (hue and saturation) features extraction is obtained directly without an intermediate transformation to *RGB* (red, green, blue) format. It is achieved by introducing a new modified relation between *RGB* and *YUV* color representations and applying subsequently the standard *RGB* – *HSV* transformation proposed by Rogers [4].

The modified $R^1G^1B^1 - YUV$ transformation is obtained as follows:

$$R^1 = U$$

$$G^1 = 217 - 0.51 * U - 0.19 * V \qquad (1)$$

$$B^1 = V$$

The algorithm developed for segmentation uses marking the blobs by specific code values in terms of the following conditions:

1. Each object defined by pixels color attributes into the limited sector from the chromaticity plane is single in the image. All the pixels, which color attributes, are outside the sector are background for this object.
2. Each object envelopes an area limited from the contour and could be erased by changing the code to the fixed one (considered for total background) and be background for the next searched object.
3. The contour finding could be combined with a single pixels and lines filtration.

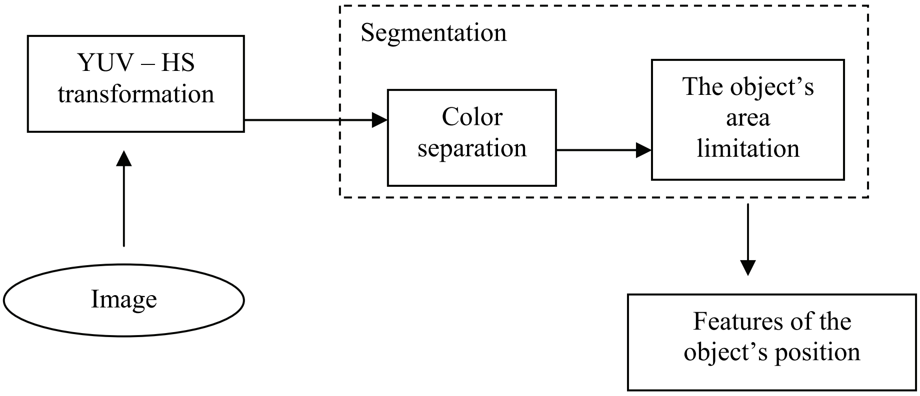
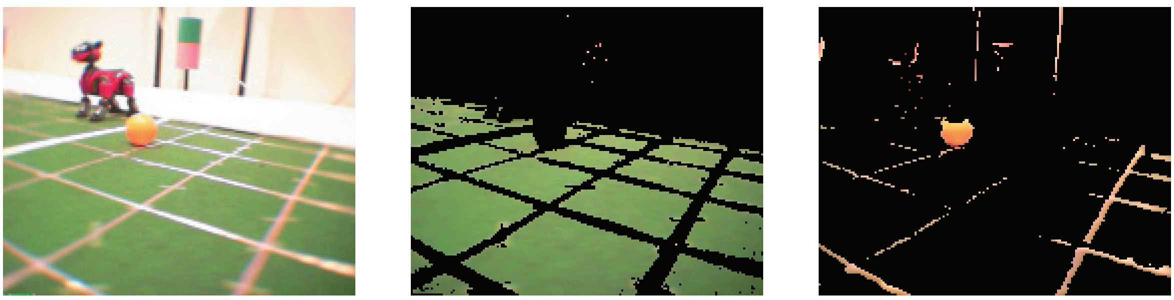
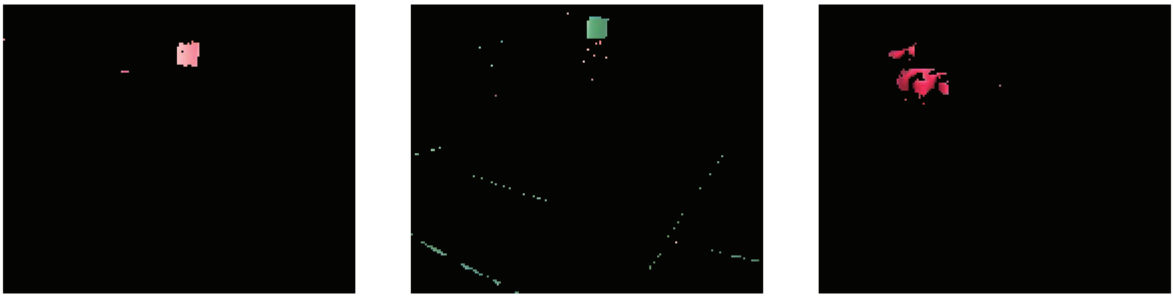


Fig. 2

The first step in the segmentation process is object's separation based on its color hue. The obtained experimental data showed that the most of important elements have a saturation value greater than 26%. Color separation insists on limiting of the ranges on the color hue scale. Figure 3 demonstrates the input image and the separated objects.
Each pixel receives a code value that relates it to a specific object. The algorithm ensures less than 10 ms operating time and it is presented in details in [5].
The next implemented image processing operation performs limiting of the contour of the objects. The border pixels are fixed by gradient operators and using a scanning mask of 3x3 pixels. The method of contour searching is completed with a filtering. The blob's first pixel is discovered by its code value. After segmentation the blob is eliminated from the image (fig.4). Contour coordinates are temporally saved in an array that helps finding the frame borders localizing the object. The size of frame gives information about the size of object (the distance to the object) and the code value informs about their type. The mass center locates to center of the frame. All the coordinates are defined in a coordinate system of the current image. The above information together with the data obtained from the proximity sensor and the gyroscope are further used for updating the local and global maps maintained by each robot.

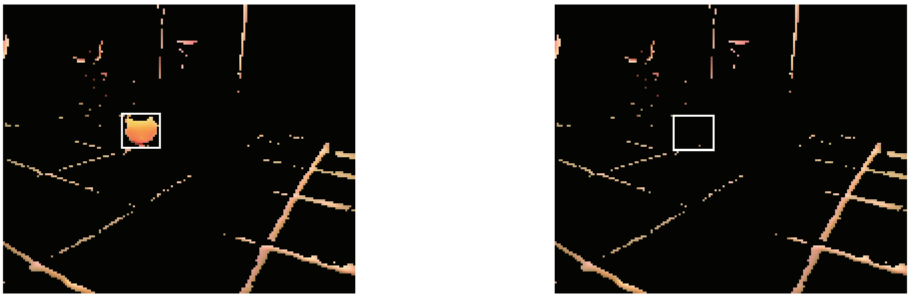


original image $50 < H < 119$ $7 < H < 46$



332 < H < 359 and Y<64 121 < H < 179 332 < H < 359

Fig. 3



frame object elimination

Fig. 4

Planning and Behaviors

The Planner module is responsible for the decision-making process. It receives data about the visible objects on the playground from the Vision module and in accordance with this information chooses the appropriate behavior. The Behaviors module consists of several basic reactive behaviors that can be switched by the Planner in order to achieve more complex behaviors of the robotic players on the field (attacker, defender, and goalkeeper). Figure 5 shows the block diagram of the applied strategy for the attacker player.

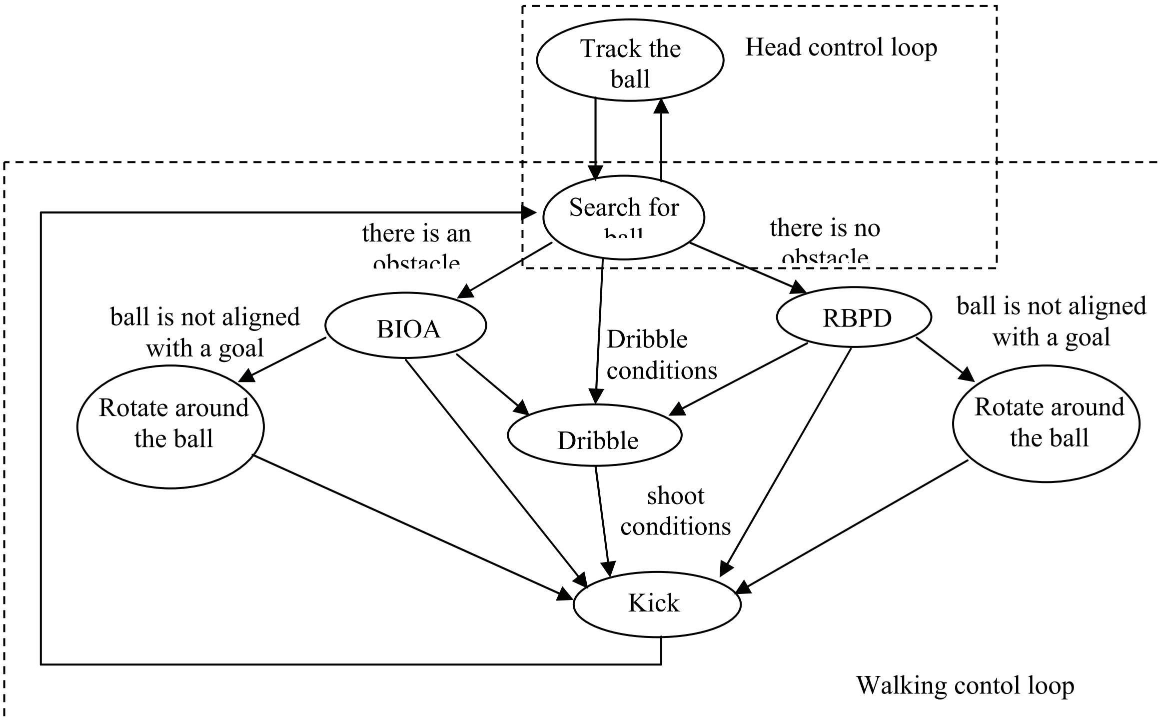


Fig. 5

Table 1 presents the list of the currently supported behaviors.

Table1

N	Description of behaviors	Needed data
1	Standing	
2	Search for the ball	
3	Track the ball	Relative angle to the ball - θ_b
4	Dribble the ball at distance	Distance – D.
5	Reaching Ball from Predefined Direction (RBPD)	Distance to the ball - D_b ; relative angle to the ball - θ_b ; angle between current direction and final desired direction - θ_d
6	Ball Interception and Obstacle Avoidance (BIOA)	Distance to the ball - D_b ; distance to the nearest obstacle - D_o ; relative angle to the ball - θ_b ; relative angle to the nearest obstacle - θ_o.
7	Kick the ball	Kick type
8	Goal-keeper standing	The type of posture
9	Rotates around the ball	Distance to ball - D_b ; Relative angle to the ball - θ_b

Dividing the Implementation level to a locomotion part and a behavior part allowed us to write a high-level control rules and behaviors that are relatively independent from the low-level control of the robot. The developed Locomotion module is hardware-dependent, and it is specifically designed for the AIBO legged robots. The behavior part is hardware-independent and can be easily ported to and from other physical platforms including also wheeled mobile robots. The type of the selected behavior and the required information about the environment are the input data for the Behavior module. The behavior outputs include linear velocities along the principal and lateral axes of the robot and an angular velocity required for implementation of the robot locomotion.
Some of the behaviors are relatively simple. The simplest one is making the robot just to stand with a predetermined posture. Others like Ball Interception and Obstacle Avoidance (BIOA) behavior and Reaching the Ball from Predefined Direction (RBPD) behavior are more complex and can be classified as intelligent behaviors. The BIOA behavior has been built as a fuzzy-neural network (figure 6) [6] - [8].

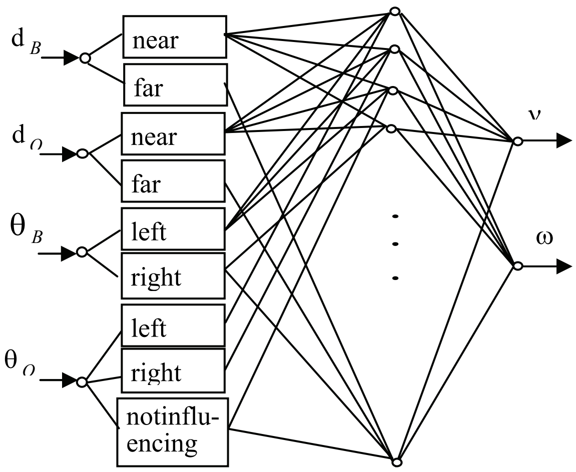


Fig. 6

The entire number of the network tunable parameters is 31. A genetic algorithm was used for tuning of the network weights. The robot reaches the moving ball avoiding the obstacles. Having as input data the current relative position of the ball and of the nearest obstacle, the BIOA behavior generates the actual linear and angular velocities for the robot locomotion. By repeating the procedure for every sample time interval a trajectory will be generated that will lead the robot to the ball. When the ball becomes closer, the robot is slowing down its velocity and finally it stops when reaching the ball.

The Reaching Ball from a Predefined Direction (RBPD) behavior aims to optimize the walking trajectory including in one behavior two tasks: to reach the ball and to turn around the ball so as a result the robot has to stop at the desired side of the ball facing to it. Doing these two tasks simultaneously the walking trajectory becomes shorter, smoother and allows higher velocities so the time for executing is much shorter. The RBPD behavior has been built as a two-layered neural network with 10 nodes in the hidden layer. It uses input data that include only the distance to the ball - D_b, relative angle to the ball - θ_b and angle between current direction and final desired direction - θ_d. The result is the linear and angular velocities of the robot. Figure 7 shows a trajectory generated by RBPD behavior.

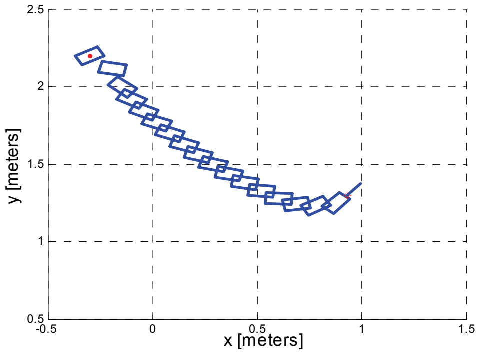


Fig. 7

Locomotion

The Locomotion module provides the interface to the physical motion functionality of the robot. The football game requires multiple different moves, but we have used only the basic ones. These movements are enough to make the dog look like a real football player. There are three main groups of movements: walking, kicking and head movements. These commands are transformed into motion by controlling 17 joints (3 for the head, 4x3 for the legs and 2 for the tail).

Head motion commands: The control of head movements is fully independent from the walking. Head motion commands include a set of “scan” commands, “look-at” command that points the head to a specific direction and “track” command, which solves the classical problem “object tracking“. These commands are implemented directly by giving appropriate set-points for the position of the three joints, which rotate the head around three perpendicular axes.

Walking: The main task, which has to be decided, is to provide stable walking and a possibility for control of the robot’s speed and direction. For these purposes an efficient gait has been developed. In every moment the robot is standing on the playground with at least three legs. It moves its legs one by one in the following order: front left, rear right, front right and rear left.

Let us choose Ox and Oy to be respectively the principal and the lateral axes of the robot and Oz to be a vertical axis. Figure 9 shows the time diagram illustrating the vertical movements

of the legs. Figure 10 shows the profile of the single step. Let us assume that this is the first step of the front left leg in accordance with the time diagram on figure 9 the part AB corresponds to first sample time (0<t<1), BC corresponds to sample time 2, CD to sample time 3 and DA corresponds to sample times 4 to 12. This gait provides an acceptable balance between the velocity of walking and stability, which is necessary for a camera's work.
The control scheme uses the inverse kinematical model of the robot to obtain the desired joint positions. PID-controllers are used to minimize the error between the desired and current joint positions. The linear locomotion velocity of the robot can be controlled by changing the size of the steps. The necessary angular velocity is achieved by using different linear velocities to right and left legs. The developed Locomotion module accepts commands from the upper control level (the Behavior module) containing three parameters: linear velocities along the principal and lateral axes of the robot and the angular velocity and translates them to a walk.
Kicking commands: Since the dogs are not just runners but also football players a set of specialized kicking routines for attackers and some postures for the goalkeeper have been also developed.

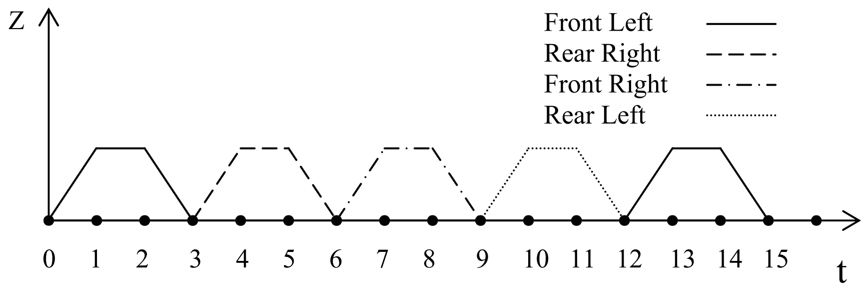


Fig. 9

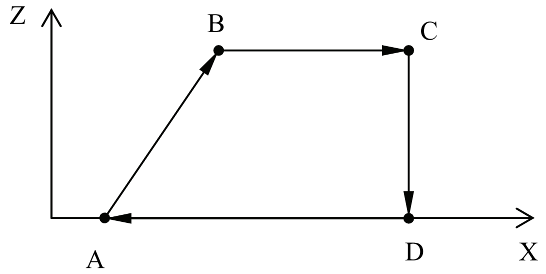


Fig. 10

DISCUSSION AND CONCLUSIONS

Robot soccer is an entertaining domain providing a large spectrum of interesting research topics. The players are entirely autonomous robotic agents that achieve complex high-level goals in real-time environments. In this paper we introduce our research efforts to build a robust and efficient system for data acquisition and control, which provide reasonable behaviors on the playground. The architecture of the control system developed for the legged quadruped AIBO robot is presented. It is designed for participating in the Sony quadruped league within the RoboCup competition.

REFERENCES

1 http://www.robocup.org
2 Bruce J., T. Balch, T., Veloso, M. Fast and Cheap Color Image Segmentation for Interactive Robots, 2000, http://www.coral.cs.cmu/fastivision.html
3 Gonzalez R., R. Woods. Digital Image Processing. Addison Wesley Publishing corp. 1992.
4 Rogers, D. Procedural elements for computer graphics. (1989) (transl.) Moskow , Mir.
5 Pavlova, P. E., Tombakov, J. K., Shakev, N. G. (2002) Fast Image Segmentation on the Basis of Color Hue and Saturation. International conference "CompSysTech'2002", Sofia.
6 Topalov, A. V. and Tzafestas S. G. (2000) Layered Multi-agent Reactive-behavior Learning in Robotic Soccer. Proc. of the 6th IFAC Symposium on Robot Control, SYROCCO'00, Vienna, September, pp.325-330.
7 Topalov, A. V. and Tzafestas S. G. (2001) Fuzzy-Neural-Genetic Layered multi-Agent reactive Control of Robotic Soccer. Chapter 18 in the book Data Mining for Design and Manufacturing. Methods and Applications., edited by D. Braha, Kluwer, Dordrecht, pp.417-442.
8 Shakev, N. G., Topalov, A. V., Kaynak, O., and Akin, L. (2001) Reactive-behavior Learning of Quadruped Robots Playing Soccer. Proc. of the 1st International Conference on Information Technology in Mechatronics, ITM 2001, October 1-3, Istanbul, Turkey, pp.149-154.