BRocks 2010 Team Description [*]

M. Akar, Ö. F. Varol, F. İleri, H. Esen, R. S. Kuzu and A. Yurdakurban [**]

Boğaziçi University, Bebek, İstanbul, 34342, Turkey

Abstract. This paper gives an overview about the current state of development and upcoming changes of the team BRocks 2010. Mechanical/electrical subsystems, control and strategy (coordination) units are described in detail.

1 Introduction

Robocup SSL remains one of the most exciting competitions of Robocup, as the game is played at a quite high pace involving extremely sophisticated strategies, which is partly possible due to the centralized camera and computer systems being used.

Several issues in terms of electronics, communication and control have to be handled in order to realize a team of robots that can compete in Robocup SSL. To achieve this objective, the BRocks team have been working within the Networked & Embedded Control Systems Laboratory at the Bogazici University since 2008. Our aim is not only to participate in Robocup competitions, but also use our testbed to develop and test our hybrid, decentralized control, coordination algorithms while taking communication, networking, vision, electronics and mechanical constraints into account. Having participated in Robocup 2009 for the first time, we would like to compete in Singapore so that we can field a stronger team on our home turf in Istanbul 2011.

The BRocks team consist of both graduate (Ö. Feyza Varol, Fatih İleri, Huzeyfe Esen, Erinç Topdemir) and undergraduate (Rıdvan Salih Kuzu, Aytaç Yurdakurban, Mehmet Öğüt, Bekir Kağan) students. In the rest of the paper, the current state of BRocks robots and testbed are described in detail. In particular, not only information about existing mechanical and electrical subsystems is given but also improvements in terms of low-level control and high-level coordination algorithms is presented.

2 Mechanical Subsystem

The mechanical subsystem of our robots is similar to other Robocup designs [1, 2] in that it is equipped with four custom-built omnidirectional wheels, a

---

[*] Submitted to Qualification Evaluation for RoboCup 2010 SSL, February 15, 2010.

[**] Please address all correspondence to Prof. Mehmet Akar, Department of Electrical and Electronics Engineering, Boğaziçi University, Bebek, İstanbul, 34342, Turkey. Tel: +90 212 3596854. Fax: +90 212 2872465. E-mail: mehmet.akar@boun.edu.tr

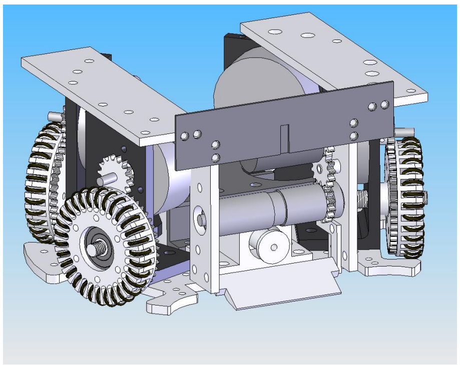

Fig. 1. Technical drawing of BRocks robots.

Fig. 2. Locomotion system: omniwheels.

dribbler and a kicking system in front. The mechanical system is the same as used in Robocup 2009. As listed in Table 1, our robots meet the mechanical specifications of the Robocup SSL.

The mechanical subsystem is composed of 3 main components (see Figs. 1–2): locomotion system, dribbler and kicker. As shown in Fig. 1, the locomotion system consists of a base and 4 omni–wheels driven by 30 watt brushless DC motors with a gear ratio of 3:1. Each of the omni–wheels consists of 30 smaller wheels wrapped around it. Both the wheels and the base of the robot were precision manufactured via CNC tools based on CAD designs.

The dribbler mechanism consists of a rotating horizontal cylinder controlled by a 6 watt brush DC motor. The rotation speed is controlled via an actuator circuit whose input comes from the main micro–controller, and it is activated once the robot has the possession of the ball. The dribbler is designed to have a ball coverage of less than 20%.

Height of the robot	143 mm
Maximum diameter of its projection onto the ground	176 mm
Maximum percentage of ball coverage	< 20%

Table 1. BROCKS Team Robots: Mechanical Specifications.

The kicker mechanism contains a push type solenoid actuated by a kicker circuit that consists of voltage amplifier and a capacitor. The associated kicker circuit is also controlled by the master micro–controller which sends the kick signal and its duration.

3 Electrical Subsystem

Our electrical subsystem follows the same structure as in the previous year. Each of our robots relies on the following electronic circuits that receive commands from the software subsystem in order to perform the desired tasks:

1. Locomotion Motor Control Circuit: Our robots consist of four custom–built omniwheels, each of which is driven by a 30 watt, 4370 rpm brushless DC motor. Two 8bit microcontrollers are used to estimate the motor speeds and a controller logic is implemented on the microprocessors for precise speed control.
2. Dribbler circuit: The dribbler consists of a 6 watt DC brush–type motor and it is driven by a simple H–bridge circuit that is controlled by the main microprocessor.
3. Kicker circuit: The design principle of our current kicker circuit is similar to other Robocup designs [1] in the sense that it relies on charging a capacitor to 160 V and then releasing the solenoid once the controlling computer sends the "kick" command.
4. Main Board: For proper implementation of the control strategies on the robots, it is critical that data be communicated to the robots in a wireless fashion that do not violate the rules of Robocup SSL. To this end, we use Zigbee low power wireless communication modules. The control data generated by the main computer are sent to the robots using the wireless modules, which are then received and processed by the microprocessors to carry out the following tasks:
   (a) Measure and control the speeds of four brushless DC motors,
   (b) Activate the solenoid when required,
   (c) Activate and control the dribbler when required.

The electrical subsystem also includes a gyroscope and an accelerometer as additional sensors to be used in order to improve the rotational motion of the robots. However, the associated control algorithms have yet to be implemented.

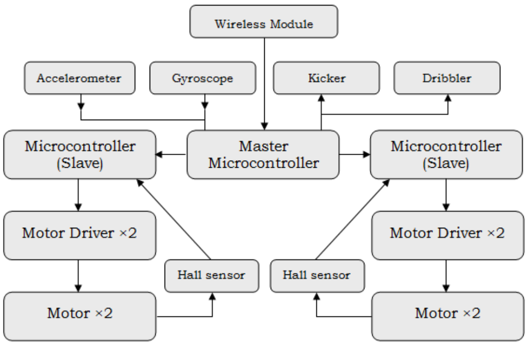

Fig. 3. The schematic of our low level control architecture.

4 Low Level Control

The schematic of our low level control architecture onboard each robot is shown in Fig. 3. The primary task of the low level control unit is to control the motor speeds. The components of the Low Level Control Module are given in Table 2. The desired motor speeds are sent to the robot via wireless Zigbee trans-receiver module from the remote PC. Microprocessors get the motor speed data from the Zigbee trans-receiver module onboard and activate the speed control loop.

Part Name	Quantity
Microprocessors	3
Wireless Zigbee trans-receiver module	1
Brushless DC motors with Hall sensors	4
Brushed DC motor with gear system for the dribbler	1
Voltage booster and charge pump circuits for the kicker	1

Table 2. Main components in the Low Level Control Module

4.1 Brushless DC Motors

Maxon EC-45 Flat 30 watt Brushless DC Motors are used for the locomotion of our robots. The main idea for choosing this type of motor is that its small size allows us to use limited space more efficiently. The motors operate with 12V, at a maximum speed of 4400 rpm and can produce 59 mNm continuous nominal torque. 1:3 gear reduction ratio is used in order to increase the overall torque and three Hall sensors with 120 degrees phase difference are available from the motors for speed measurement.

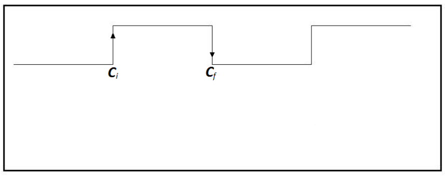

Fig. 4. PWM Signal.

4.2 Speed Estimation

Speed estimation is done using the Hall sensor outputs [3]. The period of the sensor output signal is measured with the help of a micro–controller. Then, the velocity of the motor is estimated by the conversion from electrical position to mechanical position. A counter starts to run when the micro–controller receives a rising edge from the sensor output. The counter stops when the falling edge is received. The time difference can be calculated using the difference between two values of the counter C_i and C_f which is stored for the period calculations (see Fig. 4) :

$$\Delta C = C_i - C_f. \tag{1}$$

The period of the sensor signal P_e is obtained by multiplying the reciprocal of the frequency of the counter:

$$P_e = 2\Delta C/f_c. \tag{2}$$

Since the number of pole pairs in the motor is 8, the mechanical period of the motor P_m is 8 times larger than the electrical period of one of the sensor signal, i.e.,

$$P_m = P_e \times \text{ no. of pole pairs.} \tag{3}$$

From (3), the speed of the wheel is computed as

$$V_m = 1/(8P_e) \text{ (rev/sec)} \tag{4}$$

4.3 Speed Control

The speed regulation for each wheel is achieved using a digital controller that takes the reference and the estimated speeds as inputs, and adjusts the set point into the actuator. The complete block–diagram of the digital controller is shown in Fig 5 with the variables defined in Table 3 [4].

The design of the digital controller $C(z)$ depends on identification of the actuator and motor dynamics, i.e., $G_{act}(s)$ and $G_m(s)$, respectively. The speed

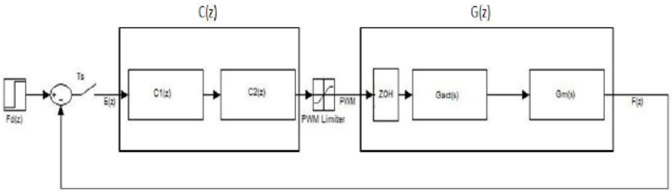

Fig. 5. Digital speed controller

$F_d(z)$	z–transform of the desired wheel frequency
$F(z)$	z–transform of the estimated wheel frequency
$C(z)$	Digital PI controller
ZOH	Zero–order–Hold
$G_{act}(s)$	Transfer function of the driver circuit
$G_m(s)$	Transfer function of the motor
T_s	Sampling period

Table 3. Descriptions of the variables in Fig. 5.

regulation is realized using a digital PI controller whose parameters are chosen such that the closed loop pulse-transfer-function is stable, and certain transient performance specifications are satisfied. For more details, see [4].

5 Vision based control and coordination

In this section, we describe the complete feedback system composed of autonomous holonomic robots that are equipped with wireless communication devices, two overhead cameras that can provide feedback on the robot positions, and a host computer that acts as a supervisor (see Fig. 6). The host computer receives/processes the vision data, and sends control commands to the robots accordingly. Our vision system consists of two 60 fps digital cameras which provide the visual feedback to the controlling computer.

5.1 Vision Subsystem

As it is mandatory to adapt the SSL–Vision software starting in Robocup 2010, we have integrated the program into our testbed. The SSL–Vision software provides the coordinates of the robots and the ball location via a graphical interface once colour and field calibrations are done properly based on the light intensity of the field. In our integration of the software, we have not experienced any difficulties so far.

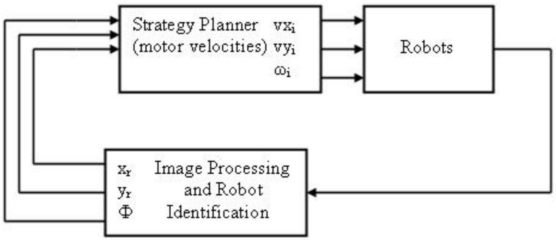

Fig. 6. Vision Based Control/Coordination Architecture.

5.2 High Level Control and Strategy Planner

High–level control of robot soccer team consists of three main modules:

1. State evaluation and mode selection: In this module, it is determined whether the team is in offensive or defensive mode.
2. Strategy planning and tactics: The strategy planning is vital in multi–robot domains. Basically, the strategy planner assigns roles to each robot in order to complete a task, e.g., scoring a goal or defending its own goal. The strategy planner mainly consists of two decision processes: a) Decision on where the robots should move to, b) Making the robots move to the desired locations.
3. Motion planning and navigation: One of the main objectives when planning paths for multiple robots is to arrive at the destination point from a given initial point, while avoiding obstacles. There are various techniques used in path planning. Frequently used techniques are classified in Table 4 [5].

Classical	*Probabilistic*	*Heuristic*
Cell Decomposition	Probabilistic Roadmaps	Artificial Neural Networks
Potential Fields	Rapidly Exploring Random Trees	Genetic Algorithms
Roadmaps	Level Set	Fuzzy Logic

Table 4. Path planning techniques

To briefly describe our methodology for the latter part, suppose that we set a goal point in the 2-D plane as shown in Fig. 7 [3, 4] . The location errors in x and y coordinates are defined as:

$$e_x = x_{goal} - x_{robot}, \tag{5}$$

$$e_y = y_{goal} - y_{robot}. \tag{6}$$

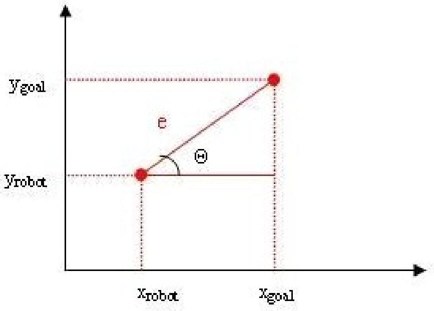

Fig. 7. Error vector definition.

Using (5–6), we create a position error vector:

$$\Theta = \tan^{-1}(e_y/e_x), \tag{7}$$

$$|e| = \sqrt{e_x^2 + e_y^2}. \tag{8}$$

In order to direct the robot towards the goal point, we need proper velocity vectors in x and y directions. To this end, we have formulated the velocities in x and y directions as follows:

$$v_x = |e| \cos \Theta, \tag{9}$$

$$v_y = |e| \sin \Theta. \tag{10}$$

The velocities are proportional to the norm of the error vector that is the distance between the desired and current location of the robot. One important thing that needs to be considered is that, calculated velocities are relative to the global coordinates. In order to have the robot motion in the desired direction, we should transform these velocities relative to the robot's current orientation. This is accomplished by using the inverse of the rotation matrix in the z direction:

$$Z^{-1}(\Theta) = Z^T = \begin{bmatrix} \cos \Theta & \sin \Theta \\ -\sin \Theta & \cos \Theta \end{bmatrix}. \tag{11}$$

Finally, the commanded velocities are calculated as

$$\begin{bmatrix} v_{xrobot} \\ v_{yrobot} \end{bmatrix} = Z^{-1}(\Phi) \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \tag{12}$$

where Φ is the orientation of the robot relative to the global coordinate system.

5.3 Path Planning

Most path planning algorithms in real–time are based on the standard path planning approach [6]. Different from last year, the approach used for path planning combines several existing algorithms. The algorithm detects obstacles from the perspective of the robot and the intended destination. It is also capable of acting in Robocup domain in real–time. The flowchart of the modified algorithm is given in Fig 8.

Multi–agent collaboration The key issue in coordinating a team of robots during an SSL game is to decompose the complex task into simpler actions which might be referred to as modes and defining the transitions between these modes in some optimal way [8]. As the constraints and the goals of SSL are known, it is a well-defined environment for developing multi-agent strategies. On the other hand, it is still a challenging test–bed since two teams of robots compete with each other to win the match. The robots should work collaboratively in order to reach success. To this end, we intend to adapt 3 different approaches in developing our multi-formation algorithms:

1. Hybrid systems based formulation and control: A hybrid system is a dynamical system whose behavior develops as the result of a continuous state system interacting with a discrete event system (See Fig. 9). We will use hybrid systems in the design of low level and high level control algorithms.
2. Market driven: The main idea of the market-driven approach is to apply the basic properties of free market economy to a team of robots in order to increase the gains of the team. In adapting the aforementioned technique to our system, we will define suitable metrics in order to select the proper actions at any given time [9].
3. Biologically inspired: In the later stages of our software development, we also plan to extend and incorporate the biologically inspired method developed in [10] to our system.

6 Concluding Remarks

Participation in Robocup 2009 for the first time has helped us improve our team significantly. We look forward to competing in Singapore so that we can field a stronger team in Istanbul 2011.

Acknowledgements

This work is supported in part by the TUBA GEBIP Programme and by the Boğaziçi University Research Fund. We also would like to thank Tekin Mericli from the Cerberus team [11] for helping us to integrate SSL Vision software and the referee box into our testbed.

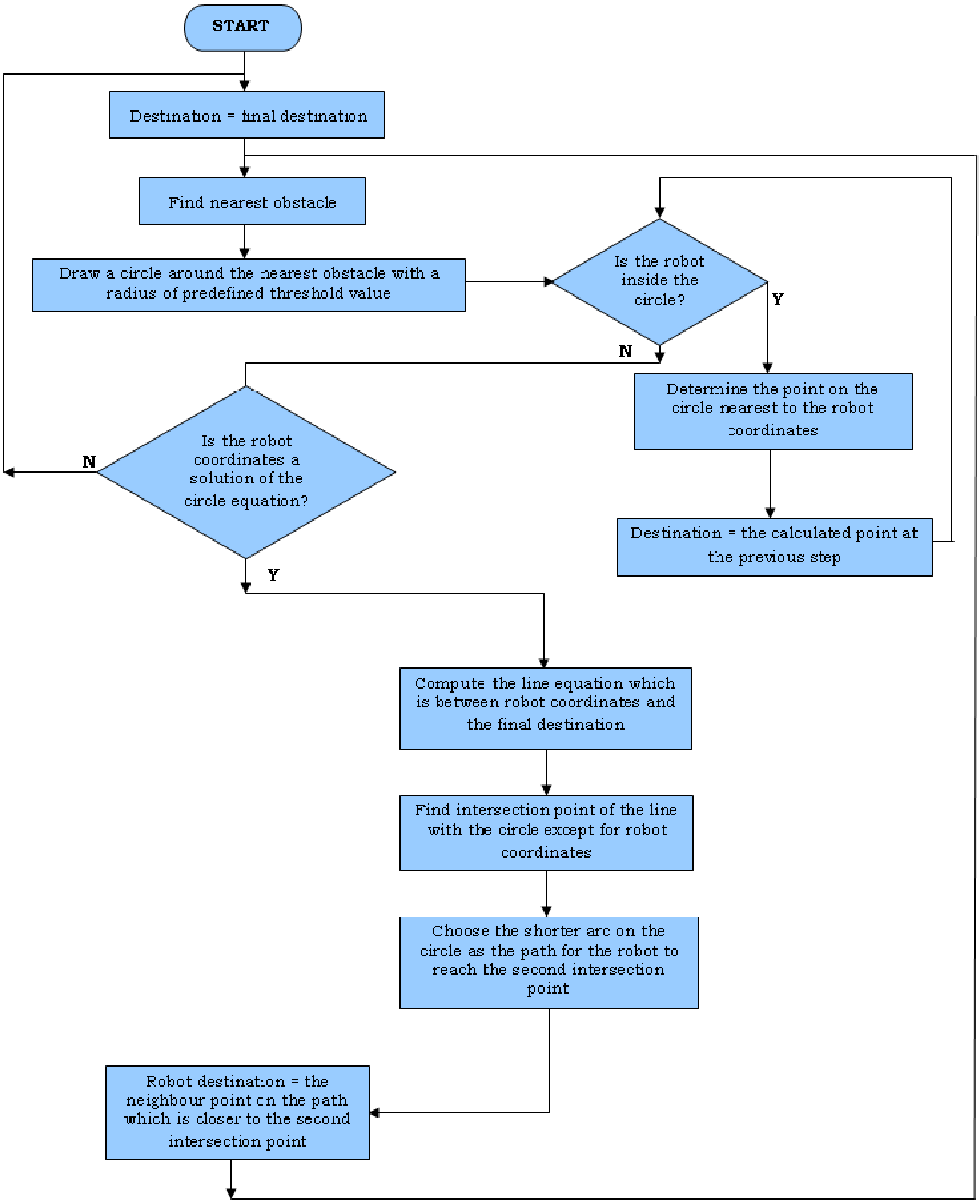

Fig. 8. Path Planning Algorithm.

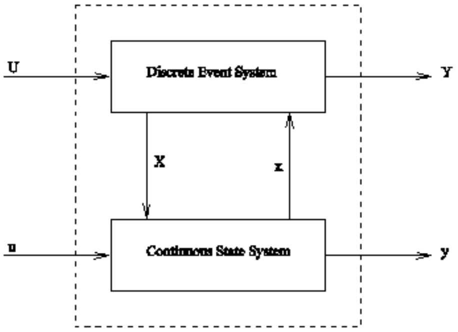

Fig. 9. Hybrid system architecture [7].

References

1. Robocup Systems Engineering Project, MS Thesis, Dept. of Electrical Engineering, Cornell University, 2002.
2. F. Wiesel, *Steuerung und Kontrolle von Omnidirektionalen Fussballrobotern*, Diplomarbeit der Informatik, Freie Universitat Berlin, January 2006.
3. H. Karaoguz, M. Usta, and M. Akar, "Design and Implementation of a Hierarchical Hybrid Controller for Holonomic Robot Formations" *Proc. of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV 2008)*, Hanoi, Vietnam December 2008.
4. H. Karaoguz, "Vision based control algorithms for a small size robot soccer team", *MS Thesis*, Bogazici University, February 2009.
5. E. Masehian, D. Sedighizadeh, "Classic and heuristic approaches in robot motion planning-a chronological review." *In: Proceedings of World Academy of Science, Engineering and Technology*, vol. 23, pp. 101-106, 2007.
6. O. Khatib, "Real–time Obstacle Avoidance for Manipulators and Mobile Robots", *The International Journal of Robotics Research*, 5(90-98), 1986
7. M. Akar, "Variations on control of hybrid systems", PhD Thesis, The Ohio State University, USA, December 1999.
8. H. Kose, K. Kaplan, C. Mericli, U. Tatlidede and H. L. Akin, "Market-Driven Multi-Agent Collaboration in Robot Soccer Domain", *in Cutting Edge Robotics*, (eds: Kordic, V.; Lazinica, A. and Merdan, M.), Germany, July 2005.
9. C. Mericli and H. L. Akin, "A Layered Metric Definition and Validation Framework for Multirobot Systems," *RoboCup International Symposium 2008*, Suzhou, China, July 15-18, 2008.
10. T. Mericli and H. L. Akin, "Soccer Without Intelligence," *IEEE ROBIO*, Bangkok, Thailand, February 2009.
11. http://robot.cmpe.boun.edu.tr/∼ cerberus/wiki/