A Distributed Multiagent Architecture for Temasek Robocup Team (TPots).

TPots

Nadir Ould Khessal, Chi Ming Kan, Sreedharan Gopalsamy, Hock Beng Lim

NO, CK, SG, HL: Temasek Polytechnic

Abstract. *Temasek Robocup team (TPots) started with the organization of the 1998 Pacific Rim Series Robocup competition in Singapore. That was our first participation to robotic soccer game competition. Since then we have been working on the development of a control architecture distributed among the physical robots and the workstation. This paper describes the overall design with an emphasis on the on board robot agents and off board agents. Robocup small league competition is considered by many AI professionals as an ideal platform for testing distributed artificial intelligence techniques. Among these techniques are multiagent systems, since its debut in the late 80's it has been used in several multi robot systems. Ranging from cellular robot systems (Fukuda et al) [1] to a team of trash-collecting robots (Arkin et al) [2]. In this paper we will present a distributed approach in implementing the multiagents architecture of Temasek Robocup team.*

1 Introduction

The major characteristic of the RoboCup soccer competition is the dynamic nature of the environment in which robots operate. The only static object in the competition field is the field itself. Team and opponent robots as well as the ball can be placed anywhere in the field, be it a purposeful strategic positioning, a missed action or a forced displacement. This has led many researchers to shift from the traditional model-based top down control to a reactive behavior based approach. Robots need not waste a huge amount of resources building maps and generating paths which might prove useless at the time of action. Instead robots are supposed to react to the actual changes in the environment in a simple stimulus-response manner. However due to the size limitations imposed by the competition rules, on-board vision systems proved to be complex and expensive alternative. Therefore building a complete perceptual and actuating system in an embedded processor is impractical. Due to these constraints, several researchers have turned to off-board global vision. Cameras are placed above the field from which relevant information about the field such as robot coordinates, identities and ball position is dumped to a stand-alone computer. This centralized

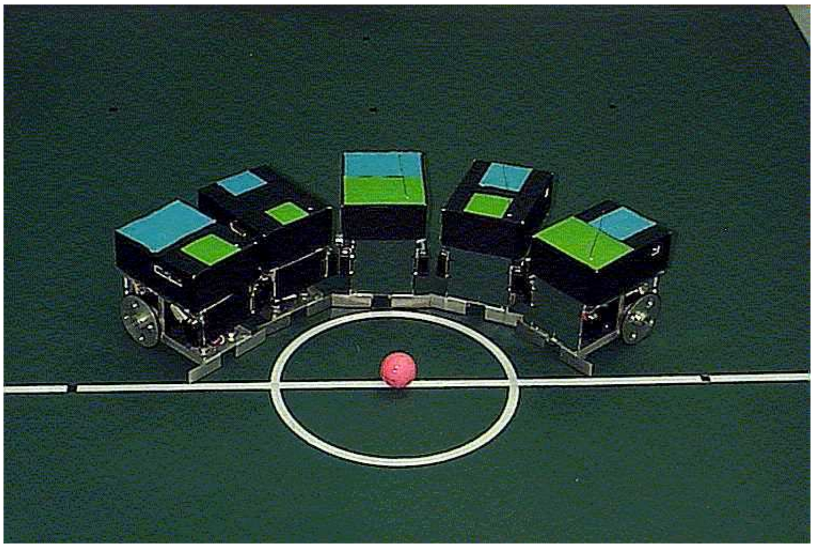

Figure 1:

approach in building the control system has led to the adoption of a hybrid (deliberative and reactive) approach. Reactive behavior based agents are embedded in the robots for urgent time-critical actions such as obstacle avoidance and command execution. Visual data manipulation and filtering as well as high-level reasoning e.g. ball position prediction, are done in the remote computer. In this paper we will present a hybrid control architecture, distributed among the robots (Figure 1) and the host computer. Associated with every robot is an embedded agent, in charge of navigating the field and executing commands generated by the remote agent. Remote agents select and implement the required tasks, based on the visual data provided by the vision system and the strategy selected by the reasoning module.

2 System Architecture

The system hardware consists of a Pentium host computer, a vision system based on Newton labs cognachrome vision card, RF transmission system and five robots. The robot :The robot on-board controller is implemented in an 8 bit processor running at 9.216MHz with an on-board memory of 512kbyte RAM and 512Kbyte EEPROM. The board also includes a real time clock and programmable timers. Sensors: Attached to the robot are three infrared sensors mounted in the front and rear to detect obstacles whilst moving. The communication module: The host computer transmits commands to the robot via radio transceivers utilizing UHF radio waves. Each robot has its own transceiver and a unique node address. The low-powered wireless system transmits less than 1mw of power and is effective over distances of 3 to 30 meters. Two-way communication rates of up to 38.4Kbps are possible. The command set is transmitted as text code piggy-backing on the transmission protocol. Commands are sent and received from the transceiver using an RS-232 interface. Vision: A global vision system, which consists of color camcorders and a special image processor (MC68332), is used. The system is able to segment and track the robots and ball at a high frame rate. Each robot has two color pads. The image

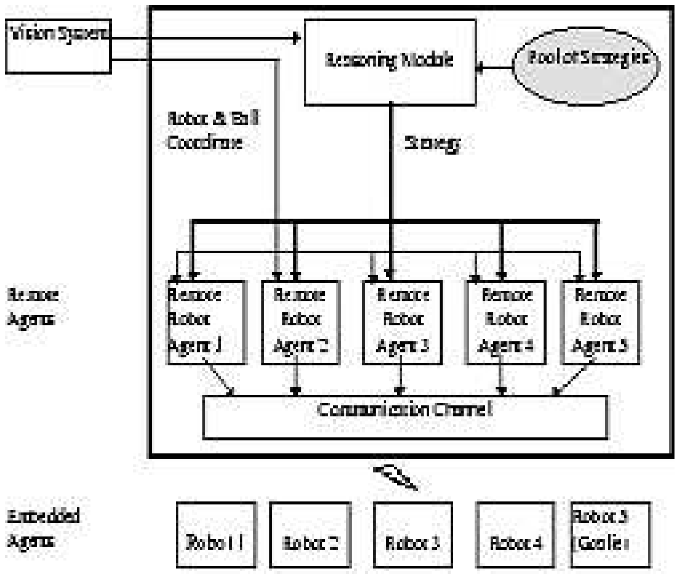

Figure 2. A Distributed Multiagent Architecture

Figure 2:

processor is trained to see the different colors and gives the locations of the center of gravity of the two color pads. Hence the orientation and robot position are known. Color pad areas are used to distinguish between different robots and minimize latency.

3 Distributed Multiagent System

Our approach in implementing the control architecture of the robots is based on dividing each robot controller into two parts: Embedded agent running in the on-board processor and Remote agent running in the off-board host computer. The embedded agent consists of several reactive behaviors competing with each other through the use of activation levels (inhibition and suppression). The main role of the embedded agent is to execute commands issued by the remote agent and avoid other robots and obstacles. The remote agent on the other hand implements strategies generated by the reasoning module. Based on the current score and performance of the opponent team, the reasoning module selects a strategy from a pool of a pre-designed strategies and downloads it to the remote agents. This enables the agents to select the appropriate tasks for each robot. Such tasks will enable the robot to intercept the ball, follow a target or simply move to a predetermined position. These behaviors are implemented in every robots remote agent except the goalie. This allows the robots to swap roles e.g. from being a defender to a forward and vice versa.

3.1 Embedded Agents

An embedded agent consists of reactive behaviors designed to perform low level navigational tasks. These behaviors are simple stimuli-response machines where the stimulus can be a sensory input or an incoming RF command. The arbitration mechanism is based on a fixed prioritization network[3]. The response of a single higher priority behavior takes over the control of the robot whenever the associated stimulus is present (Figure 3). Obstacle avoidance is the main autonomous task done by the robot, using the three infrared sensors, the robot uses the two machines i.e. Avoid Left and Avoid Right to move away from the obstacle by turning to either left or right and moving a certain distance until the straight line path is clear. The remote agent detects that the robot is out of the previous ly computed path and re-computes and transmits a new path. Due to the dynamic nature of the environment, obstacles are not taken in consideration while planning a path for the robot.

Remote agents transmit paths in the form of a turning angle followed by a traveling distance. The two machines on the robot i.e. Turn and Move are in charge of executing these commands. Note here that unlike the Move machine, the Turn Machine is of a highest priority and therefore uninterruptible.

3.2 Remote Agents

For the robot to be able to play soccer it needs some basic skills such as moving towards the ball position, kicking the ball towards the goal area, intercepting the ball and passing the ball to a team member. Most of these skills could be performed by the following behaviors: - Intercept_ball: This machine enables the robot to move behind a predicted ball position before kicking it towards a target area. The target area could be the opponent goal keeper area (in an attempt to score a goal (Figure 4.), a clear area in front of a team member (ball passing) or simply the opposite side of the field, in the case of a defending robot. This behavior is achieved by first moving to an intermediate position. Once there the robot charges towards the ball. The intermediate position is determined by computing the two lines starting at the edges of the target area and intersecting at the predicted ball position (L1 and L2 in Figure 4). The distance between the intermediate point and the ball predicted position d is fixed (charging distance). The intermediate position corresponds to the midpoint of the base segment of the equilateral triangle b.

- Follow: This machine is designed to keep the robot following a target object. The target can be the ball, a team robot or an opponent robot. This is done to keep the robot nearer to the ball and therefore in a better position to intercept the ball. - Homing : Depending on the strategy being executed, robots could be required to move to a certain position for the purpose of forming a defense wall for example. Such actions are performed by the Homing machine.

Each of the above machines contains a path planner. This planner generates a path in the form of a straight line between the robot and its target position. After the path has been generated and transmitted the planner keeps track of the robot position. This will enable the remote agent to detect divergence of the robot from its most recent path. Divergence of the robot could be caused by a hit from an opponent robot or purposeful move to avoid an obstacle . In both cases the path planner computes and transmits a new path to the robot.

4 Future Work and Conclusion

After our first participation in the 1998 RoboCup Pacific Rim Series, Small League [4], several changes related to the vision system and RF communication have been made. These changes have enhanced considerably the system performance and reliability. Currently efforts are underway to use genetic algorithms in implementing a stand-alone training module for the robots. Although the technique does not guarantee an optimal global solution, it can produce reasonable solutions under certain defined conditions.

5 Bibliography

[1] Fukuda, T. et al. Structure Decision for Self Organizing Robots Based on Cell Structures- CEBOT. In Proceeding of the IEEE International Conference on Robotics and Automation. Los Alamitos, Calif, 1998.

[2] Arkin, Ronald. C, Tucker R. Balch. Cooperative Multiagent Robotic Systems. In Artificial Intelligence and Mobile Robots, ed. David Kortenkamp,278-296. Cambridge, Mass.: The MIT Press.1998.

[3] Nadir Ould Khessal. An Autonomous Robot Control Architecture Based on A hybrid behavior arbitration. In The Proceedings of Modeling , Identification and Control Conference. Innsbruck, Austria 1999.

[4] Nadir Ould-Khessal, Dominic Kan, Sreedharan Gopalasamy, Lim Hock Beng. Temasek Polytechnic RoboCup Pacific Rim Series Small League Team. In Proceedings of RoboCup Workshop.5th Pacific Rim International Conference On Artificial Intelligence. ed. Hiroaki Kitano, Gerald Seet. Singapore, 1998.