CMU Hammerheads Team Description

Rosemary Emery, Tucker Balch, Jim Bruce, Scott Lenser, Mikhail Osterfeld, Rande Shern, Kevin Sikorski, Ashley Stroupe, John Sweeney, and Manuela Veloso

Robotics Institute, Carnegie Mellon University
Pittsburgh, PA, 15213
{remery,trb,jbruce,slenser,rande,jds,veloso}@cs.cmu.edu
{mo2u,kws,astroupe}@andrew.cmu.edu
http://www.cs.cmu.edu/~coral/minnow

Abstract. In this paper the hardware and software design of the CMU Hammerhead middle-size robot team are presented. The team consists of 4 fully autonomous robots with wireless communication and color vision. TeamBots, a Java-based software package, forms the robots' software architecture, while Clay, a TeamBots package for creating motor-schema based control systems, is used for developing behaviors. TeamBots provides a rich multi-robot simulation environment that allows concurrent development of behaviors in simulation and their refinement on the mobile robots.

1 Introduction

The CMU MultiRobot Lab is focused on the study of teams of robots that operate in dynamic and uncertain environments. In order to accomplish its goals, the group has been dedicated to the construction of inexpensive, autonomous robots that can be used for its research. The CMU Hammerheads represents CMU's first entry into the RoboCup middle-size league and the team is the result of the MultiRobot Lab's efforts.

The domain of robot soccer is a very challenging one; not only does it require each team member to possess a variety of capabilities, they must also be able to work together to achieve goals. Behaviors necessary to soccer such as tracking the ball have been identified and are being developed as a set of one or more sub-behaviors depending on their complexity. Low level behaviors are selected by a finite-state machine to produce the overall behavior of a team member. Communication between players allows for the development of a world model that will aid in selecting between behaviors. The information contained in this model augments the sensory information gathered through the camera and bump sensor with which each robot is equipped.

This paper describes the CMU Hammerhead team. Section 2 outlines the hardware design of the team and its vision system. The software and control architectures used are presented in Section 4. Finally, the overall strategy and skills of the team are explained in Section 5.

2 Robot Design

The CMU Hammerheads, as shown in Figure 1, are designed to be inexpensive, reliable, fully autonomous robots with wireless communication and color vision. Onboard control is provided by Java-based software running on a Linux microcomputer. The underlying mechanical platform consists of Probotics' commercially available Cye robot [1]. This platform was chosen for a variety of reasons. It is relatively inexpensive, has very good odometry and is able to tell when it bumps into an obstacle. The platform is composed of two parts: a differentially steered motorized drive unit and a trailer. The trailer serves as a mount for the on-board computer, vision system and power supply. The total cost of each robot is just under three thousand dollars.

The on-board computer is a Pentium MMX processor-based single board computer with audio, VGA/LCD and Ethernet. The computer is mounted on the trailer. The drive unit is commanded via a 19.2k serial protocol. The computer is powered through a 12v sealed lead acid battery. Each robot is networked using a wireless technology to communicate with both the other Hammerheads and other computers. This allows for easy debugging and control of the robot.

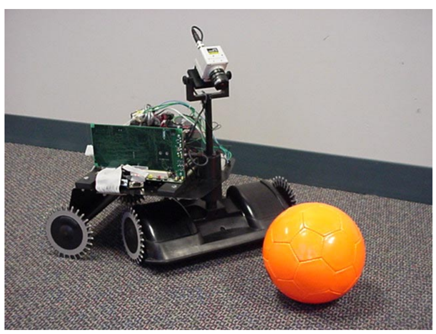

Fig. 1. A member of the CMU Hammerhead team.

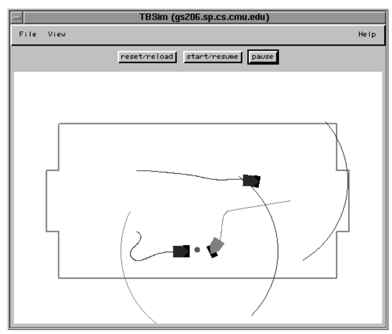

Fig. 2. TeamBots simulation environment showing multiple robots engaged in a two-against-one behavior.

2.1 Vision System

While the Cye's bump sensor is very reliable, in a game of soccer it is also essential for players to be able to proactively avoid collisions with other players as well as locate the ball and goals. For this reason the Hammerheads employ a color vision system as their primary mode of information gathering about the world. The robots each have a miniature color video camera which interfaces with the computer through a video capture card to provide 160x120 images at 30Hz. This camera is mounted over the center of the robot's drive unit and attached such that the robot can scan the field by staying in place and rotating its drive unit. The camera's tilt is fixed at an angle that allows both the robot to see a portion of itself as well as the horizon.

Images captured by the video card are processed by CMVision [2] to detect regions of pre-specified colors. CMVision is a simple and robust vision system that performs low level blob detection at 30Hz. This package provides classification and segmentation of several hundred regions of up to 32 colors. Colors are specified by defining a region in YUV space that encapsulates that color and possible variations due to lighting conditions. The results of the color blob detection are then used by the control system of the robot to detect the ball, goals and other players, and to guide its behavior.

3 Software Architecture

The CMU Hammerheads are controlled using TeamBots [3], a Java-based system that runs on the on-board computer. TeamBots is a collection of application programs and Java packages for multiagent robotics research. The strength of TeamBots is that it supports protyping in simulation of the exact same control systems that are run on mobile robots. The simulation environment, shown in Figure 2, allows for multiple robots, each running a different control system, to interact. Thus the Hammerhead team behaviors can be developed and tested in an extremely flexible simulation environment and the identical control systems can then be used on the mobile robots. Development of behaviors such as localization and ball handling can be developed in simulation while the mobile robots are being constructed.

For every robot type supported by TeamBots there exists an abstract interface to that robot's hardware. There are then simulated and hardware implementations of that interface. Any control system developed for the robot can therefore be used on both the simulated and mobile robot, as the interface is identical for both. The interaction between a control system and the appropriate robot implementation is handled by manager that controls overall operation (see Figure 3). Under this manager, the control system and robot implementation effectively run in parallel with commands generated by the control system being passed to the robot and any information requested by the control system passed back. The robot will continue to act upon the last commands sent to it by the control system until the control system generates new ones.

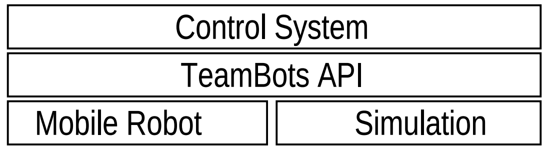

Fig. 3. CMU Hammerhead software architecture.

3.1 Control Architecture

All control systems used with the CMU Hammerheads make use of the Clay architecture [4], which is a group of Java classes that are easily combined to create motor-schema based control systems. These systems can either be simple, reactive systems or more complex hierarchical configurations that take advantage of learning and memory.

At the basis of Clay are perceptual, motor-schema and action nodes that are combined within the Hammerhead control system as behavioral sequences [5]. Perceptual nodes take information from the vision system and bump sensor as to the location of objects of interest or obstacles. These perceptual nodes are fed into motor-schema nodes that produce vectors representing the desired trajectory of the robot. Action nodes are then formed by the combination of motor-schemas. For example a dribbling action node might be the combination of an avoid-obstacles motor-schema, a push-ball motor-schema and a go-to-goal motor schema. Action nodes are the basis of a finite-state machine with each action node representing a state and perceptual nodes dictating the links between states. This system translates to low-level behaviors being represented at the motor-schema level while compound behaviors are represented at the action node level.

The overall output of the control system is a vector that represents a heading that the robot should travel along and a velocity with which to travel. The robot implementation then translates that heading and velocity command into wheel speeds as appropriate.

Simple base behaviors have been identified and corresponding motor-schema nodes are currently being developed. Some of these base behaviors include: avoiding obstacles, moving to the ball, moving to the goal, kicking the ball, moving with the ball, passing the ball, and blocking the ball. Using a finite-state machine these behaviors blend together to produce the overall behavior of a team member. Due to Teambot's multi-robot simulation capabilities, more complex team behaviors such as passing can be developed while testing of the more basic behaviors takes place on the mobile robots.

4 Strategy and Skills

The CMU Hammerheads will have a fixed goalie and three additional team members that, while starting out in designated roles, will be able to float between being defenders and attackers. The goalie role is a fixed role in order to prevent misunderstanding of which robot is responsible for defending goal in the case of a communication failure. The other roles are loose in that a robot will take on the role of an attacker or defender based upon its current position and the position of other players and the ball. This is used rather than fixing the boundaries in which the robot can act in order to allow a 'defending' robot to take advantage of a position that allows it to score.

4.1 Communicaton

The CMU Hammerheads will be taking full advantage of communication between team members to help coordinate team efforts such as passing and blocking and to allow each robot to maintain a world model. This world model will be common to team members and will include information as to the ball's location, the location of teammates and opponents as well as to the uncertainty as to each of these locations. The model will be used by the robot to determine if it should engage in attacker or defender type of behaviors.

In addition, despite the excellent odometry provided by the Cye platform, localization that uses detection of landmarks such as goals and field corners, and world model information will be used during game play.

References

1. Cye. URL:http://www.personalrobots.com, 2000.
2. J. Bruce, T. Balch and M. Veloso. Fast and cheap color vision on commodity hardware. *Workshop on Interactive Robotics and Entertainment*, to appear.
3. TeamBots. URL:http://www.teambots.org, 2000.
4. T. Balch. Clay: Integrating motor schemas and reinforcement learning. College of Computing Tech Report GIT-CC-97-11, Georgia Institute of Technology, Atlanta, Georgia, March 1997.
5. R. Arkin and D. MacKenzie. Temporal coordination of perceptual algorithms for mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 10(33):276–286, June 1994.